Technical Report 852

MANPRINT Methods Monograph: Aiding the Development of Manned System Performance Criteria

Jonathan D. Kaplan, Editor
U.S. Army Research Institute

Susan Dahl, Ron Laughery, and Rick Archer
MicroAnalysis and Design, Inc.

Lawrence O'Brien
Dynamics Research Corp.

Edward Connelly
Performance Measurement Associates

Joseph Conroy III
Science Applications International Corp.

W. Peter Cherry, David Thompson, Lolt Proegler, Dennis Roerty, and David Roberts
Vector Research Corp.

June 1989

United States Army Research Institute
for the Behavioral and Social Sciences

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION
Unclassified

1b. RESTRICTIVE MARKINGS
--

2a. SECURITY CLASSIFICATION AUTHORITY
--

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE
--

3. DISTRIBUTION/AVAILABILITY OF REPORT
Approved for public release; distribution is unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
--

5. MONITORING ORGANIZATION REPORT NUMBER(S)
ARI Technical Report 852

6a. NAME OF PERFORMING ORGANIZATION
Dynamics Research Corporation (DRC), (continued)

6b. OFFICE SYMBOL *(If applicable)*
--

7a. NAME OF MONITORING ORGANIZATION
U.S. Army Research Institute for the Behavioral and Social Sciences

6c. ADDRESS *(City, State, and ZIP Code)*
DRC--Wilmington, MA (continued)

7b. ADDRESS *(City, State, and ZIP Code)*
5001 Eisenhower Avenue
Alexandria, VA 22333-5600

8a. NAME OF FUNDING/SPONSORING ORGANIZATION
U.S. Army Research Institute for the Behavioral and Social Sciences

8b. OFFICE SYMBOL *(If applicable)*
PERI-SM

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
MDA903-86-C-0412
MDA903-86-C-0414, MDA903-86-C-0416

8c. ADDRESS *(City, State, and ZIP Code)*
5001 Eisenhower Avenue
Alexandria, VA 22333-5600

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO
62722A	791	121	C2

11. TITLE *(Include Security Classification)*
MANPRINT Methods Monograph: Aiding the Development of Manned System Performance Criteria

12. PERSONAL AUTHOR(S)
Kaplan, Jonathan D. (Editor, ARI) (continued)

13a. TYPE OF REPORT
Final

13b. TIME COVERED
FROM 86/11 TO 87/05

14. DATE OF REPORT *(Year, Month, Day)*
1989, June

15. PAGE COUNT
262

16. SUPPLEMENTARY NOTATION
Contracting Officer's Representative: Christine Hartel

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*

MANPRINT	ROC
Criteria	Specification
Performance	RAM

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

The U.S. Army Research Institute (ARI) is conducting a program to develop methods to successfully integrate available operations and maintenance personnel with weapon system hardware and software. This monograph consists of three papers that describe alternate concepts for the development of rigorous operations and maintenance performance criteria on a per system basis.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL
Jonathan D. Kaplan

22b. TELEPHONE *(Include Area Code)*
(202) 274-8923

22c. OFFICE SYMBOL
PERI-SM

Previous editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

U.S. ARMY RESEARCH INSTITUTE
FOR THE BEHAVIORAL AND SOCIAL SCIENCES

A Field Operating Agency Under the Jurisdiction
of the Deputy Chief of Staff for Personnel

EDGAR M. JOHNSON **Technical Director**	**JON W. BLADES** **COL, IN** **Commanding**

Research accomplished under contract
for the Department of the Army

Dynamics Research Corporation
MicroAnalysis and Design
Performance Measurement Associates
Science Applications International Corporation
Vector Research Corporation

Technical review by

John D. Fletcher
Christine Hartel

A-1

NOTICES

DISTRIBUTION: Primary distribution of this report has been made by ARI. Please address correspondence concerning distribution of reports to: U.S. Army Research Institute for the Behavioral and Social Sciences, ATTN: PERI-POX, 5001 Eisenhower Ave., Alexandria, Virginia 22333-5600

FINAL DISPOSITION: This report may be destroyed when it is no longer needed. Please do not return it to the U.S. Army Research Institute for the Behavioral and Social Sciences.

NOTE: The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

ARI Technical Report 852

6a. NAME OF PERFORMING ORGANIZATION (continued)
MicroAnalysis and Design (MA&D),
Performance Measurement Associates (PMA),
Science Applications International Corporation (SAIC),
Vector Research Corporation.

6c. ADDRESS (continued)
MA&D--Boulder, CO,
PMA--Reston, VA,
SAIC--McLean, VA,
Vector Research Corporation--Ann Arbor, MI.

12. PERSONAL AUTHOR(S) (continued)
O'Brien, Lawrence; (DPC),
Dahl, Susan; Laughery, Ron, and Archer, Rick; (MA&D),
Connelly, Edward; (PMA),
Conroy, Joseph III; (SAIC),
Cherry, W. Peter; Thompson, David; Proegler, Lolt; Roerty, Dennis; and
Roberts, David; (Vector Research Corporation).

Technical Report 852

MANPRINT Methods Monograph: Aiding the Development of Manned System Performance Criteria

Jonathan D. Kaplan, Editor
U.S. Army Research Institute

Susan Dahl, Ron Laughery, and Rick Archer
MicroAnalysis and Design, Inc.

Lawrence O'Brien
Dynamics Research Corp.

Edward Connelly
Performance Measurement Associates

Joseph Conroy III
Science Applications International Corp.

W. Peter Cherry, David Thompson, Lolt Proegler, Dennis Roerty, and David Roberts
Vector Research Corp.

Manned Systems Group
John F. Hayes, Chief

Systems Research Laboratory
Robin L. Keesee, Director

U.S. Army Research Institute for the Behavioral and Social Sciences
5001 Eisenhower Avenue, Alexandria, Virginia 22333-5600

Office, Deputy Chief of Staff for Personnel
Department of the Army

June 1989

Army Project Number
2Q162722A791

Manpower, Personnel, and Training

Approved for public release; distribution is unlimited.

FOREWORD

Products are only as good as the concepts that underlie them. The U.S. Army Research Institute for the Behavioral and Social Sciences (ARI) is conducting a program to develop MANPRINT methods to successfully integrate available Army personnel with weapon system hardware and software. This program is the result of a set of detailed concepts defined by ARI.

This monograph describes three alternative concepts for building a method to develop rigorous operations and maintenance performance criteria for manned systems. These concepts will serve as the focus of current system building and may serve as a seed bed for the development of alternative systems.

EDGAR M. JOHNSON
Technical Director

6 blank

MANPRINT METHODS MONOGRAPH: AIDING THE DEVELOPMENT OF MANNED SYSTEM PERFORMANCE CRITERIA

EXECUTIVE SUMMARY

This monograph consists of three papers on a common subject: The development of complete, rigorous, and operationally measurable performance criteria for manned systems. Each of these papers presents a concept for building an aiding method.

The U.S. Army Research Institute for the Behavioral and Social Sciences ~~(ARI)~~ began the program to develop methods to integrate available operations and maintenance personnel with hardware and software. The first stage of this process was to develop three alternate, competitive concepts for each method.

The three concept papers in this monograph were written in response to requirements for a method to develop rigorous and ultimately measurable performance criteria. These criteria would enable hardware and software designers to better understand what a manned, fully integrated system would have to do to achieve operations and maintenance success. Success would be described in terms of required performance levels of operations and maintenance tasks under specified conditions.

The concept papers written in response to this requirement have three significantly different focuses and bring powerful but different approaches to the problem of developing rigorous and meaningful performance criteria. Ultimately, the ARI study advisory group decided to implement the concept proposed by Micro-Analysis and Design.

MANPRINT METHODS MONOGRAPH: AIDING THE DEVELOPMENT OF MANNED SYSTEM PERFORMANCE CRITERIA

Jonathan D. Kaplan, Editor
U.S. Army Research Institute

CONTENTS

MANPRINT MFTHODS MONOGRAPH: AIDING THE DEVELOPMENT OF MANNED SYSTEM PERFORMANCE CRITERIA

Introduction

The U.S. Army Research Institute (ARI) is conducting a research program to develop methods to aid in successfully integrating available operations and maintenance personnel with hardware and software as part of the general MANPRINT process. To do this, ARI defined and produced requirements for six classes of aiding methods. The first four of these methods will aid the integration process by developing information that will be used as system design constraints. This information will be used in requirements documents and will be provided to potential design organizations. The last two of these methods will aid the integration process by providing mechanisms to evaluate system designs.

This monograph consists of three papers on a common subject: The development of operations and maintenance performance criteria that will constrain system design. Each of these three papers presents a concept for building an aiding method for developing these criteria. These methods, if built, would all be software based and provide aid without significantly raising their user's workload.

To fully understand these requirements, one must first understand the context in which they were developed. To develop information that leads to successful manning, first one must understand that "successful" implies a system that is capable of reaching some desired performance level that has been designated as a success. Second, one must understand that manning implies integration of hardware, software and soldiers into a system. To reach required performance levels, those performance levels must be determined and communicated to designers in a clear fashion. It is difficult to design a system that is capable of performing some actions well enough to reach some goals if you do not know what the actions are, or what "well enough" means, and you have only a fuzzy idea of what the goals are.

The notion of success has several attributes that must be understood and worked with. These include:
(1) description--the words that describe performance;
(2) qualification--the words that describe the variables that affect performance; (3) relationship--the hierarchical linkage among descriptions of performance, and (4) value--the minimally acceptable level of described performance in the presence of sets of variables (or conditions). Level of performance can be described in terms of duration and accuracy dimensions. Typically, these two dimension must be

linked for either to have any useful meaning. Any method for developing a definition of successful (or criterion) performance requires that the existence of each of these four attributes be understood and dealt with.

This monograph was driven by a requirement to develop alternate concepts for developing rigorous system performance criteria. The three concepts presented were written by personnel from MicroAnalysis and Design (MA&D) and Dynamics Research Corp. (DRC), Science Application International Corporation (SAIC) and Performance Measurement Associates (PMA) working as subcontractors to Applied Science Corporation, and Vector Research Corporation. Eventually, ARI chose to complete the development of MA&D's concept. However, all three concepts have considerable merit and are quite diverse in approach.

MA&D-DRC provided separate concepts for developing criteria for operations and maintenance. Their operations concept is based on the use of system level simulation models. In this concept the user inputs a mission level criterion and runs an existing or modified model of a system of the appropriate type. The model computes mission performance based on its subfunction data and compares it with the mission criterion. The user adjusts the subfunction data and reruns the model, as required, until the mission criterion is achieved. The maximum times and minimum accuracies resulting in the mission criterion are established as subfunction criteria.

The maintenance concept developed by MA&D-DRC is based on the use of a spreadsheet approach. The user selects appropriate subsystems. A system availability criterion is entered and the spreadsheet automatically alters maintenance task times at the subsystem level to those required to achieve the availability criterion, holding constant the relationships among maintenance times. The user examines the resulting new maintenance times, and if any are unacceptable he alters them and reruns this spreadsheet in the opposite direction to compute resulting availability. This process continues until criterion availability is achieved with maintenance times that are acceptable.

The Vector Research concept combines a mechanism for aiding the top-down, hierarchical decomposition of performance from the theater to the system level with information on how to develop criterion values and where to find important data for this purpose. It can be thought of as an intelligent guide and assistant. It is based on a series of production rules that contact generic data. It does not differentiate the development of operations and maintenance criteria.

The SAIC-PMA concept may be thought of as a method for eliciting objectives and criteria from experts who did not realize they knew them. It is based on optimal control theory. The user is presented with various alternate scenarios that are present in a database. Based on these scenarios, experts make judgements of expected effectiveness of the unit of which an individual system is a part. The mechanism for capturing these judgements is an anchored scale. After many judgments are obtained, multiple regression is used to identify the set of performance descriptors that are significantly related to unit effectiveness scores. Those descriptors become the system's performance criteria. The same approach is used both for operations and maintenance criteria.

The three concept papers in this monograph have been paginated as E1, E2, and E3 to delineate them clearly.

MANPRINT METHODS MONOGRAPH:

AIDING THE DEVELOPMENT OF MANNED SYSTEM PERFORMANCE CRITERIA

SYSTEM PERFORMANCE REQUIREMENTS ESTIMATION AID

Written by:

Susan Dahl
Ron Laughery
Rick Archer
MicroAnalysis and Design

Lawrence O'Brien
Dynamics Research Corporation

CONTENTS

Page

CONTENTS (Continued)

Page

CONTENTS (Continued)

CONTENTS (Continued)

CONTENTS (Continued)

Page

LIST OF TABLES

LIST OF FIGURES

SYSTEM PERFORMANCE REQUIREMENTS ESTIMATION AID

SECTION 1 - INTRODUCTION

Objective of Paper:

This concept paper describes an aid for systematically estimating system performance requirements for Army weapon systems during the earliest phases of the acquisition process. This aid is one of six products being developed in the Army Research Institute's (ARI) MANPRINT Methods development program.

The concept paper descɪ**ibes requirements for this aid and** presents a detailed description of the aid's steps and the techniques for developing them. It also outlines an approach for implementing the aid.

The concept paper is the first step of a three-step development process. In the second step, we will develop detailed design specifications. In the third step, software, documentation, and training for the aid will be produced.

Overview of the Approach:

We have named Product 1 the System Performance Requirements Estimation Aid (SPREA). This aid will help an analyst establish functional and task performance criteria for missions that result in the system attaining the minimally acceptable system performance, as established by appropriate battlefield combat models. In this summary, we will briefly explain the steps that the analyst will take in using the SPREA to accomplish this task.

First, the analyst will use the SPREA Mission Library to **specify the mission that the system is expected to accomplish.** This Mission Library will contain a variety of mission statements for a variety of systems. The analyst will be able to use one of these statements, modify an existing statement to fit his or her needs, or specify a new mission from scratch.

The analyst will also specify the minimally acceptable system performance requirements for that mission. The analyst will get these requirements from combat models.

Next, the analyst will use the Function and Task Libraries that are in the SPREA to specify the composite functions and tasks of the mission. The Task Library will also contain "baseline" values for the performance criteria for each task. These baselines will have been gathered from the analysis of comparable fielded systems or expert judgement. The task performance criteria will include data for: 1) task performance time, 2) task performance accuracy, and 3) task performance reliability.

The SPREA will use the task performance criteria to build a model of the mission which the system is expected to accomplish. This model will be executed, and the output will include predicted system mission performance. These predictions will be based on the individual task performance criteria. Next, these predictions will be compared to the minimally acceptable performance criteria, dictated by the appropriate combat models. If the predictions do not meet the minimally acceptable performance, then the individual task performance criteria must be altered to resolve the deficiencies. This resolution will be accomplished by the SPREA, employing backsolving techniques, under the direction of the analyst.

This process will iterate until all of the deficiencies have been resolved, or until the analyst interrupts the SPREA to stop the process.

The SPREA Final Report will document the explicit mission statement that was modeled, its composite functions and tasks, and the performance criteria for each of the tasks. Also, the report will include a listing of the system performance criteria that were gathered. This list will compare the predicted system performance to the minimally acceptable system performance which was an output of the combat models.

Finally, the report will include formatted input for four materiel acquisition documents, including the Letter of Agreement (LOA), the Operations and Organizational (O&O) Plan, the Required Operational Capability (ROC) and the Justification for Major System New Start (JMSNS).

SECTION 2 - PRODUCT REQUIREMENTS

2.1 Objectives

The purpose of this product is to provide Army personnel and combat developers with a tool for systematically describing performance requirements for major weapon systems. These performance requirements must be derived from system mission requirements and linked to the battlefield combat models. In addition, the System Performance Requirements Estimation Aid (SPREA) should permit the estimation and analysis of objectively defined measures of system performance under a variety of environmental, operational, and tactical conditions. The tool will not distinguish who or what performs certain functions. Rather it will determine what the functions and their associated performance standards must be under given conditions in order to achieve the required overall system performance.

2.2 Major Outputs

The primary outputs of Product 1 are lists of missions, functions, tasks, conditions, and performance requirements for a major weapon system. Missions describe the purpose(s) for which the system is being built. These missions are broken down into lower level functions and tasks before allocation among hardware, software, and humans takes place. The conditions describe the variety of situations under which missions might be performed. Performance requirements describe both the type and level of performance required for each system task under specified conditions.

2.3 Role of Product Output in Acquisition Process

The output of Product 1 will feed several key acquisition documents. This section identifies those documents and indicates how the output is presented (i.e., format), where the information is obtained, and who is responsible for the document. There are two basic types of documents that describe system performance requirements -- Army requirements documents designed to provide guidance to the Army organizations in charge of system development and contractor specifications designed to provide detailed guidance for the contractor who is developing the system. There should be a close relationship between these two types of documents. In fact, the contractor specifications should be derived from the Army requirements documents.

2.3.1 Army Requirements Documents

The primary Army requirements documents into which the SPREA will feed are the Justification for Major System New Start (JMSNS), the Operations and Organizational (O&O) Plan, the Letter of Agreement (LOA), and the Required Operational Capability (ROC). The JMSNS, O&O Plan, and the LOA should be produced during the Requirements Technology Base Activities Phase of the Materiel Acquisition Process (MAP). The ROC should be produced during the Proof-of-Principle Phase.

The requirements documents described above (JMSNS, O&O Plan, LOA, and ROC) are typically prepared by the Directorate of Combat Development (DCD) within the proponent TRADOC schools in close coordination with the Army Materiel Command (AMC) proponent.

Appendix B contains formats for the requirements documents. More specific details on the relationship between the requirements documents and SPREA output lists are presented in the sections that follow.

Missions. The JMSNS is typically the first document developed to describe the need for a major weapon system. It is a very high level document that must not exceed three pages. The JMSNS identifies mission areas but does not necessarily describe the missions.

TRADOC PAM 70-2 details the O&O Plan and states that the operational plan section must "describe how, what, when and where the system will be employed on the battlefield and how it will interface with other systems" (p. 3.6). It also requires that an Operational Mode Summary (OMS) be attached as an annex. Figure 1 displays an example mission profile from the TRADOC PAM 72-2, the Materiel Acquisition Handbook. As the example indicates, the guidance for describing missions is vague. The OMS is also used as the mechanism for describing missions in the ROC and LOA.

Functions and Tasks. The O&O Plan, ROC, and LOA all require that the OMS "reflect tasks analyzed in the Mission Area Analysis" that preceded the MAP. The regulations governing these documents provide little additional guidance on the nature of these functions and tasks. However, additional guidance is provided in the regulations (TRADOC PAM 11-8, Appendix C) developed for the mission area analysis (MAA) process. This guidance recommends that a hierarchy of tasks and subtasks be developed. A preliminary list of tasks is included in Appendix D.

LETTER OF AGREEMENT (LOA)

SAMPLE MISSION PROFILE
IFV MISSION PROFILE (U)

1. OPERATIONAL ENVIRONMENT. (+ High; 0 Med; - Low)

THREAT	LEVEL	ENVIRONMENT	LEVEL
Artillery	+	Day	+
Tank Main Gun	+	Night	+
Small Caliber Gun	+	EW	0
AD Gun	-	Smoke	+
Ground Launched ATGM	+	Haze	0
Air Delivery ATGM	0	Fog	0
High Performance Aircraft		0 Dust	0
Mines	-	Built-up Areas	0
		Rain	+
		Sleet/Snow	0

2. MISSION. (Percent of time.)

ATTACK	DEFENSE	TOTAL
40	60	100

3. TYPE TARGET. (Percent of total targets expected to be engaged by system, 1986.)

Tanks	35
Lightly Armored Vehicles	40
AD/SP Artillery Guns	12
Bunkers	4
Helicopters	5
Personnel	4

4. ENGAGEMENT RANGE DISTRIBUTION. (Percent of total targets expected to be engaged by range band.)

RANGE (Meters)	ATGM	AUTOMATIC CANNON
0 - 500	10	30
500 - 1000	40	30
1000 - 2000	40	25
2000 - 3000	10	15

Note: This is not the actual IFV Mission Profile.

Figure 1. Sample Mission Profile.

Once functional tasks are identified, the proponent analyzes each task and subtask to determine how each contributes to the mission's success. The analysis is done "using subjective but sound military judgment" (TRADOC PAM 11-8, p. C-11).

Conditions. TRADOC PAM 11-2 states that a brief paragraph in the O&O Plan and the O&O summary in the LOA and ROC must indicate:

a. How the equipment will be used;

b. Geographical areas of use;

c. Weather and climatological factors to be considered during equipment operations;

d. Battlefield conditions (such as ECM, smoke, and dust) in which the system will operate; and

e. The type of units that will use and support the equipment.

The OMS Annex of the O&O Plan, LOA, and ROC also provide lists of conditions.

Minimally Acceptable Performance Requirements. Performance requirements are used in the O&O Plan to help:

> Describe the need for an operational capability to defeat the threat and eliminate an operational deficiency . . . (The need should be stated in broad characteristics only (e.g., a capability is needed to defeat enemy armor at "x" kilometers)). (TRADOC PAM 70-2, p. 3-6).

The LOA and ROC require that the performance characteristics of a proposed system be described in bands of performance. The lower level of these bands should describe minimally acceptable performance. (TRADOC PAM 70-2, p. 5-17, 6-12, 7-13).

AR 702-3 provides specific guidance on the metrics that can be used to describe RAM requirements. The SPREA will consider a subset of these metrics as described in the Software Specifications Section of this concept paper.

2.3.2 Documents for Presenting Requirements to Contractors

While the Army requirements documents described above define system performance requirements for Army organizations, these documents are typically not the primary mechanism used to present requirements to contractors. The Army requirements documents may be included in the RFP package as background information, but the contractor is not contractually bound to meet the requirements in these documents. Rather, the requirements documents that the contractor must adhere to are stated in the system specification. MIL-STD-490 describes procedures for describing system specifications. The first system specification that is typically developed for a major weapon is the System Segment Specification (SSS) or Type A specification. The SSS should be initially developed during the Requirements Technology Base Activities Phase of the MAP, but may be updated in the subsequent phase. It is typically developed by the combat developer within the proponent school but may be contracted out. Data Item Description DI-CMAN-80008 describes the format for the SSS.

Compared to the Army requirements documents, the SSS Data Item Description (DID) provides much guidance on the format for describing system functions and tasks and performance requirements. But it provides little guidance for describing missions and conditions. Table 1 lists the relevant sections that describes SPREA output.

It is interesting to note that the SSS includes a place for identifying system effectiveness models.

> 10.2.5.2.11 System Effectiveness Models. This subparagraph shall be numbered 3.2.11 and specify the requirements to develop system effectiveness models. In addition, this subparagraph shall specify the level of detail to which each system effectiveness model shall be developed.

The SSS also includes a section on qualification requirements. This section describes the methods to be used to show that each system performance requirement has been met.

In later MAP phases, more detailed system specifications are developed. However, these specifications require an allocation of functions among particular system steps, and they actually describe requirements at the step level. Consequently, these specifications are not relevant to the SPREA.

Table 1. Guidelines for Describing SPREA Products from SSS DID (DI-CMAN-80008).

Missions

10.2.5.1.1 Missions. This subparagraph shall be numbered 3.1.1 and describe the missions of the system.

Functions/Tasks

10.2.5.1.3 System Modes and States. This subparagraph shall be numbered 3.1.3. If the system can exist in various modes, this subparagraph shall specify each mode and provide a brief description of each mode (e.g., weapon idle, weapon readiness, weapon deployment). In addition, if the system can exist in various states, this subparagraph shall specify each state and provide a brief description of each state (e.g., surveillance, threat evaluation, weapon assignment, target designation and acquisition, fire control resolution). If applicable, this subparagraph may also reference a system mode/state table to specify the correlation between system modes and states.

10.2.5.1.4 System Functions. This subparagraph shall be numbered 3.1.4 and divided into the following subparagraphs to describe each system function.

10.2.5.1.4.1 (Name X) System Function. This subparagraph shall be numbered 3.1.4.X (beginning with 3.1.4.1), specify function X by name and number, and describe its purpose. This subparagraph shall also identify and define the applicable parameters within function X (e.g., input, output) and specify its performance and physical characteristics. Functional flow and schematic diagrams may be used to identify and define the applicable parameters within function X. Performance and physical characteristics shall include requirements allocated from system requirements as well as requirements which are peculiar to the function and cannot be directly referenced to system requirements. If applicable, this subparagraph shall specify the modes and states in which function X operates.

10.2.5.1.5 System Functional Relationships. This subparagraph shall be numbered 3.1.5 and describe the top-level functional relationships between system functions (i.e., chronological, high-level control, etc.). This description may be provided by a system functional flow diagram (see Figure 1). This subparagraph shall also define the functional and physical interfaces between all system functions.

Conditions

10.2.5.1.2 Threat. This subparagraph shall be numbered 3.1.2 and contain the following:

a. The model or characteristics of potential targets.

b. Characteristics of the current and potential enemy weapon capabilities.

c. Any additional threat considerations that affect the system design.

10.2.5.2.2 Environmental Conditions. This subparagraph shall be numbered 3.2.2 and specify the environmental conditions to be encountered during the transportation, storage, and operation of the system. The following conditions shall be specified:

a. Natural environment (e.g., wind, rain, temperature).

b. Induced environment (e.g., motion, shock, noise).

Table 1. Guidelines for Describing SPREA Products from SSS DID (DI-CMAN-80008) (Continued).

c. Electromagnetic signal environment.

d. Shipboard magnetic environment.

e. Environments due to enemy action (e.g., over-pressure, blast, underwater explosions, radiation).

10.2.5.2.10 Deployment Requirements. This subparagraph shall be numbered 3.2.10 and specify the anticipated deployment of the system both geographically and organizationally (e.g., the number of installations and their operating locations).

Performance Requirements

10.2.5.1.6.1.1 Functional and Performance Requirements. This subparagraph shall be numbered 3.1.6.X.1 (beginning with 3.1.6.1.1) and identify by name and number the function(s) that HWCI (or) CSCI X performs. In addition, this subparagraph shall specify the associated performance requirements. (e.g., "The Weapons Interface Unit CSCI shall perform the following functions:

a. Function 1, Weapons Update - update a weapon's data base every 2 seconds In addition this update shall require not more than 350 MS and be capable of updating at least 8 missiles in the data base within the 350 MS.

b. Function 2, etc.").

10.2.5 4.1 Reliability. This subparagraph shall be numbered 3.4.1, specify reliability requirements in quantitative terms, and define the conditions under which the reliability requirements are to be met This subparagraph may include the reliability apportionment model to support apportionment of reliability values assigned to system functions for their share in achieving desired system reliability

10.2.5 4.2.1 Maintainability. This subparagraph shall be numbered 3.4.2.1 and specify quantitative maintainability requirements. The requirements shall apply to maintenance in the planned maintenance and support environment and shall be stated in quantitative terms. Examples are

a. Mean and maximum down time, reaction time, turnaround time, mean and maximum times to repair, mean time between maintenance actions.

b. Maximum effort required-to locate and fix an error.

c. Maintenance man-hours per flying hour, maintenance man-hours per specific maintenance action, operational ready rate, maintenance man-hours per operating hour, frequency of preventative maintenance.

d. Number of people and skill levels, variety of support equipment.

e. Maintenance costs per operating hour, man-hours per overhaul.

10.2.5.4.3 Availability. This subparagraph shall be numbered 3.4.3 and specify the degree to which the system shall be in an operable and committable state at the start of the mission(s), where the mission(s) is called for at an unknown (random) point in time.

2.4 Users

2.4.1 Overview of Users and Their Functions

Primary Users. The primary SPREA users will be the combat developers within the TRADOC proponent schools who produce requirements documents for major systems (i.e., JMSNS, O&O Plan, LOA, and ROC) and who produce the SSS which guides early contractor design activities. The organization which typically accomplishes these functions is the Directorate of Combat Development (DCD). Within DCD, portions of the requirements documents and SSS may be completed by a Concepts and Studies Division, Materiel Logistics Support Division, or Requirements Division. Each DCD is organized slightly differently.

When detailed specifications for the SPREA are developed, we will identify the specific DCD organizations within each major TRADOC proponent. This will be accomplished by examining the AR 10 series for each school to identify the organization specifically assigned the responsibilities for producing requirements documents and SSS.

Secondary Users. Another major user is expected to be the AMC major subordinate command who may provide input to the TRADOC combat developer in constructing requirements documents. Since each AMC major subordinate command is organized differently, the exact user organization will vary. Again during development of detailed design specifications for the SPREA we will use the AR 10 series to develop a detailed list of specific organizations. Typically, the AMC command will have an Advanced System Directorate (ASD) with a Requirements Analysis Division (RAD) responsible for coordinating requirements documents with TRADOC.

Other potential users are the reviewers of requirements documents which include HQ TRADOC (DCSCD), HQ AMC (AMCDRE), and the Requirements Division (DAMO-FOR) within DCSOPS; the MANPRINT Policy Office within ODCSPER (DAPE-ZAM); the MANPRINT points-of-contact within the TRADOC proponent schools and AMC subordinate command; and the ARI field office representatives who may provide MANPRINT support to TRADOC schools or AMC subordinate commands.

2.4.2 Job Type

TheSPREA will be specifically developed for the primary users listed above -- that is the combat developers within the TRADOC proponent school who produce requirements documents for major systems (i.e., JMSNS, O&O Plan, LOA, and ROC) and who produce SSS. The individuals who actually perform these functions within the assigned DCD division are typically military at the rank of major or captain. We will develop a more definitive list of job types during the development of detailed design specifications for the SPREA. This will be accomplished by contacting the appropriate division for DCDs for the major TRADOC proponent schools.

2.4.3 Additional Information on Users

During the development of the detailed design specifications, we will gather additional information on (1) user training background and (2) current and projected hardware and software available to users. We will develop this information by contacting the appropriate division for DCDs for the major TRADOC proponent schools.

2.5 Assumptions

The assumptions underlying development of the SPREA are:

Major System Focus

The SPREA will be developed to describe system performance requirements for major weapon systems. This means that while the general logic of the SPREA could be applied to other types of systems, the automated tools in the SPREA will only be developed for major systems.

Input From Combat Models

Combat models will provide input to the SPREA on overall system performance requirements for each mission. The SPREA will aid the analyst by providing a bridge to the appropriate combat model and to the appropriate measures of effectiveness within that model. Since total system performance requirements at the mission level are, for the most part, input to the SPREA, the SPREA's primary function is to provide a tool to help analysts allocate these system level performance requirements to individual mission functions and tasks.

System Effectiveness Model

Our concept for describing the total effectiveness of a major weapon system is based on the WESIAC model described in the DARCOM-P 706-101. This concept is also congruent with the performance measurement model in Kaplan's (1986) system performance model.

According to the WESIAC model, System Effectiveness is a measure of the extent to which a system may be exprcted to achieve a set of specific mission requirements. System Effectiveness is a function of availability, reliability, and capability.

Availability is the probability that the system will be operable at the start of a mission.

Reliability is the probability that the system will complete the mission given that it was operable at the start.

Capability is the probability that the mission tasks will be performed correctly (i.e., without error).

2.6 High Level Functional Requirements and Constraints

2.6.1 Technical Requirements

Output. The SPREA will assist Army analysts in producing clear, unambiguous descriptions of system missions, functions, functional tasks, conditions, and performance requirements.

Emphasis on Functional Requirements. The SPREA will describe the minimally acceptable system performance requirements prior to the allocation of functions to hardware, software, and humans. Thus, the requirements describe what has to be done without describing the mechanism that will perform the function.

Mission and Functions. The missions and functions must be stated in clear, unambiguous terms that facilitate the development of measurable performance requirements.

Conditions. The SPREA must describe environmental, operational, and tactical conditions.

Performance Requirements. Performance requirements must be stated in clear, unambiguous, and directly measurable terms. They must describe minimal acceptable performance levels for the conditions under which the system missions must be performed.

Role In Acquisition Process. The SPREA information on missions, functions, functional tasks, conditions, and performance requirements must be designed to feed directly into Army requirements for major weapon systems (i.e. JMSNS, O&O Plan, LOA, and ROC) and the Type A specification that guides contractor designs. (See the information on Role in the Acquisition process earlier in this section.)

Users. The SPREA must be designed for the combat developers within the TRADOC proponent school who produce requirements documents for major systems (i.e., JMSNS, O&O Plan, LOA, and ROC) and who produce System Segment Specification (SSS) which guide early contractor design activities (see the Overview of Users and Their Functions.)

2.6.2 Acceptability and Usability Requirements

The previous subsection presented an overview of the technical requirements that the SPREA must meet. This section describes some of the acceptability and usability requirements which also must be met by these tools.

Produce Tailored User Outputs and Processes. Previous R&D products have not been implemented because they failed to meet the needs of individual Army decision makers. They were R&D products "in search of users". To avoid this problem in the current effort, it is critical that specific users be identified for the SPREA. Furthermore, the output of the SPREA should be formatted so that Army users can insert them directly into MAP documents. Additionally, the aids must be capable of producing results in a timely fashion and of meeting the requirements of the new streamlined acquisition process. The latter indicates a need for using some form of automation to support each product whenever it is cost effective to do so. Finally, to develop products that meet users needs, users must be involved in all phases of product development.

Describe "How To" Procedures. Sufficient "how to" procedures must be included in the SPREA to allow Army users with minimal training to use each product. Whenever possible, procedures will be automated to reduce user analysis requirements. However, for all automated tools, detailed procedures for obtaining input data and interpreting results will be presented. For all manual tools, detailed instructions for conducting each analytical step will be provided.

Minimize Organizational Impacts. The SPREA must be designed to fit the user and not vice versa. Consequently, it must not require additional personnel or cause restructuring of existing Army organizations; it must utilize computer hardware available at user locations or be accessible via secure lines.

Minimize User Training. The members of the MAP community who are expected to use the SPREA are already overburdened and understaffed. In addition, they are trying to meet increasing acquisition requirements such as MANPRINT within the context of the streamlined acquisition process. Consequently, training time for the $(MPT)^2$ products must be minimized. This requires development of user interfaces that require no prior computer experience. For example, the interface should contain built-in job aids (e.g., help commands). Finally, when formal training is required, it must be developed in accordance with Army instructional system design principles and utilize only media that are readily available or accessible to users.

Security. The SPREA may be required to accept classified data and must be designed to provide acceptable levels of security.

SECTION 3 - PRODUCT OVERVIEW

This section provides an overview of the first product of the MPT^2 project, the System Performance Requirements Estimation Aid (SPREA).

The purpose of this product is to provide Army personnel and combat developers with a tool for systematically determining system functional performance requirements based on overall system performance requirements. These overall system performance requirements will be derived from system mission requirements as determined by the battlefield combat models. The SPREA will then permit the analyst to estimate and analyze objectively defined measures of system performance under a variety of environmental, operational, and tactical conditions. The SPREA will not distinguish who or what performs each task in the mission. However it will help the analyst determine the functions and associated performance standards required under given conditions to achieve required overall system output.

In order to keep this product concept in perspective, it is necessary to state that we are following the mission, function, and task definitions from Kaplan and Crooks (1980). In this context, system missions are statements of the overall purpose(s) of the system. Missions are decomposed into functions that are the higher order activities that the system is designed to perform and that are the basis for the overall performance requirements of the system. Finally, the functions themselves are then decomposed into tasks.

A simplistic example of the decomposition process is: The mission "Transport troops" contains the functions "Navigate" and "Protect System from Threat." The function "Navigate" contains the following tasks:

"Identify present location"
"Identify destination"
"Select travel route"
"Travel designated route"

Figure 2 provides an overview of the steps involved in using the SPREA. This section presents a brief discussion of these steps. More detail is provided in Sections 4 and 5 of this concept paper.

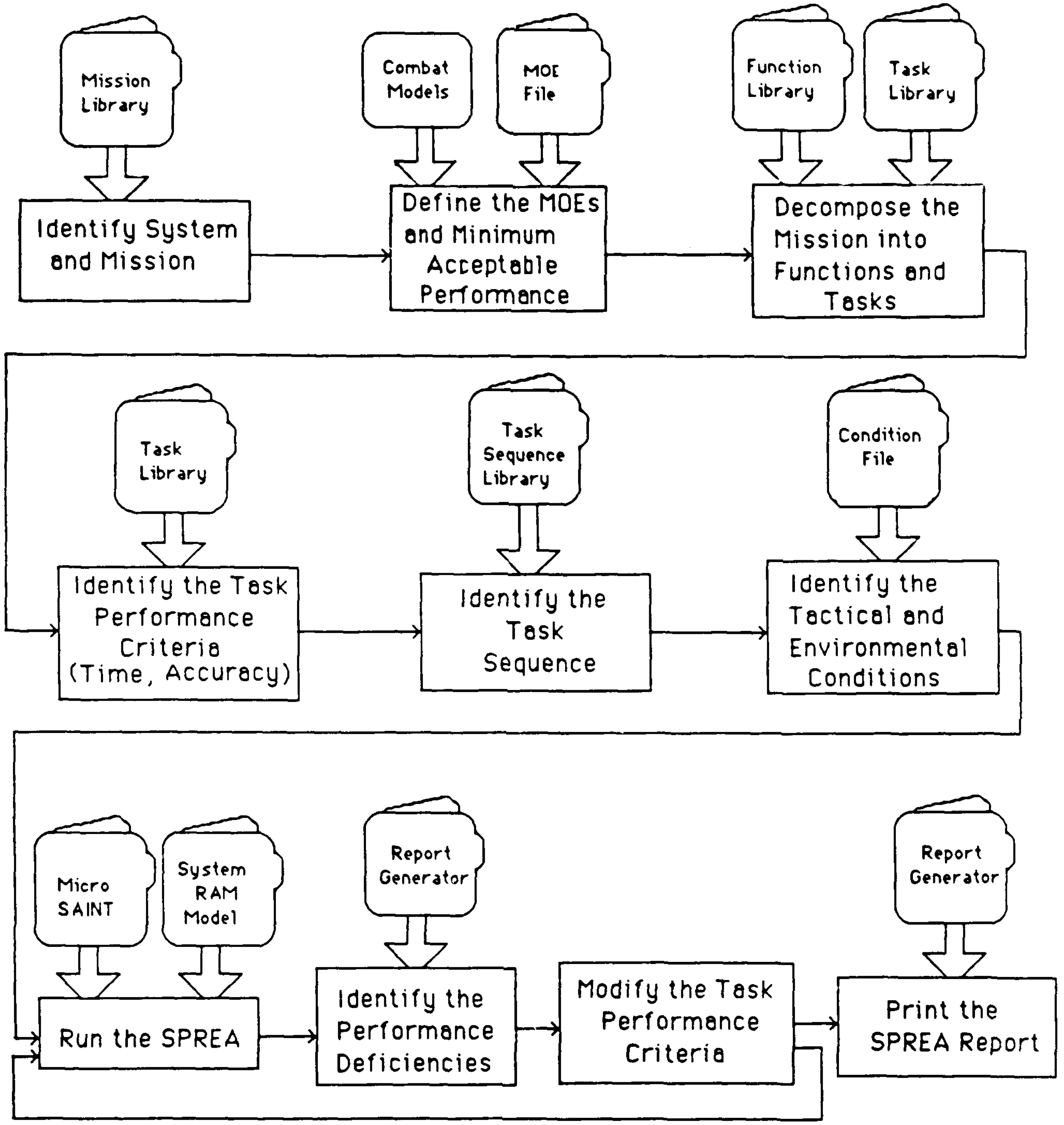

Mission Library
Identify System and Mission
Combat Models
MOE File
Define the MOEs and Minimum Acceptable Performance
Function Library
Task Library
Decompose the Mission into Functions and Tasks
Task Library
Identify the Task Performance Criteria (Time, Accuracy)
Task Sequence Library
Identify the Task Sequence
Condition File
Identify the Tactical and Environmental Conditions
Micro SAINT
System RAM Model
Run the SPREA
Report Generator
Identify the Performance Deficiencies
Modify the Task Performance Criteria
Report Generator
Print the SPREA Report

Figure 2. SPREA Steps

3.1 Outputs

The primary SPREA output will be a detailed report containing:

- an explicit statement of the missions that were modeled and their composite functions and tasks. The statement will also list the performance criteria for each of the tasks in each mission.

- the environmental conditions in which the system will operate

- the minimally acceptable system performance including:

  - a system reliability estimate (i.e., the probability of mission completion)

  - the operational availability requirement

  - a system maintainability estimate

  - a system mission capability or accuracy estimate

  - time to perform the mission

- formatted input to the Letter of Agreement (LOA), the Operations and Organizational (O&O) Plan, the Required Operational Capability (ROC) and the Justification for Major System New Start (JMSNS)

Each of these areas are briefly discussed in the following paragraphs. Section 4 of this concept paper presents a more thorough explanation.

3.1.1 Mission, Function, and Task Documentation

The SPREA Report will fully document the mission which was modeled, as well as its composite functions and tasks. This documentation will also include a spreadsheet listing the tasks with their performance criteria. These performance criteria include:

- most likely task performance time
- maximum task performance time
- accuracy of task performance
- probability of task completion (reliability estimate)

Finally, the documentation will supply a network drawing that indicates the predecessor and successor relationships between the tasks.

3.1.2 Conditions

The SPREA will remind the analyst of the task performance parameters that might be affected by changing tactical, operational, or environmental conditions. The SPREA will prompt the analyst to note which conditions he or she is assuming when setting the task performance criteria, and will include those notes for inclusion into the SPREA Report.

3.1.3 System Performance Estimates and Deviations from Requirements

The SPREA will aid the analyst to input the minimally acceptable performance that the system must meet (as derived from combat models) and the allocation of these performance requirements to individual mission functions and tasks. The SPREA Report will note any discrepancies between the minimally acceptable system performance and the system performance predicted by modeling the composite functions and tasks. The SPREA Report will also document the source of these discrepancies (i.e., the "critical path" through the times and accuracies of the tasks included in each system performance measure). The SPREA will provide backsolving techniques to assist the analyst in identifying a set of functional performance allocations that will produce minimally acceptable performance.

Mission Accuracy. The SPREA will use the task performance accuracy criteria to estimate mission accuracy. This estimate will be calculated by combining the performance accuracies for the tasks that are in the mission model. This methodology is discussed in Section 5.

Reliability. One of the performance criteria associated with each task is the probability of completing the task without failure. As the simulation progresses, the SPREA will combine these reliability estimates for each task to calculate an overall system reliability estimate. The SPREA Report will include this value.

Availability. Availability is an important piece of the performance measurement framework that guides the SPREA. There are many measures of availability; however, AR 702-3 states that operational availability (Ao) shall be used in all requirements documents. We assume that appropriate combat models will produce an availability requirement and the composite pieces of the availability equation, and the analyst will be asked to supply these values as input into the SPREA. The RAM Model which is discussed in Section 5 of this concept paper will utilize these values.

The SPREA Report will document the input values that the analyst supplies, the source of the data, and any comments the analyst includes.

Maintainability. Maintainability of the system is a measure of the time it takes to retain or restore the system to a specified operable condition. Maintainability is one of the components of availability. The SPREA will ask the analyst to input the mean (assuming an exponential distribution) of the time that it will take to maintain the system. Most likely, these data will come from maintenance data of comparable systems. The analyst may also have to solicit estimates from subject matter experts.

This maintenance time will be used in the RAM model (discussed in Section 5) and the analyst will be able to change the maintenance value in order to study the resulting availability requirement.

The SPREA Report will document the maintenance parameter input that the analyst supplied as well as the resulting RAM Model availability criteria.

In this product, the tasks have not yet been allocated to humans, software, or hardware. If the same component (e.g., computer) is being used for two or more tasks, however, then it would be necessary to represent the maintainability of those tasks as interdependent. It is not meaningful to speak of maintaining a task, it is only meaningful to maintain equipment so that we can track the amount of time that a component has been operating, etc. The lack of component allocation does not allow this, and is the primary reason that maintainability can not be represented at a task level in this product. At this point in the system requirements analysis process, we feel that the combat models are the most appropriate method of gaining a meaningful maintainability requirement. However, we are very open to suggestions and other ideas on this matter.

3.1.4 Formatted Input to Documents

In order for the SPREA Report to have optimal utility throughout the Army community, it must be in a format which feeds specific requirements documents. The Directorate of Combat Development typically prepares four documents that will receive input from the SPREA. These documents are the Justification for Major System New Start (JMSNS), the Operations and Organizational (O&O) Plan, the Letter of Agreement (LOA), and the Required Operational Capability (ROC). The format of the SPREA Report will be specifically geared to the formats of these documents. Examples of these documents are provided in Appendix B, and they were discussed in more detail in Section 2.3.1 of this concept paper.

3.2 Integration with other $(MPT)^2$ Products

The first four $(MPT)^2$ products, the System Performance Requirements Estimation Aid, the Manpower Constraints Estimation Aid, the Personnel Constraints Estimation Aid, and the Training Constraints Estimation Aid, are designed to estimate MPT-related requirements and constraints during the earliest phases of the acquisition process, the phase involving Requirements Technology Base activities. The System Performance Requirements Estimation Aid (SPREA) will assist Army combat developers in identifying comprehensive, clear, and unambiguous system performance requirements and missions.

The Manpower Constraints Estimation Aid (MCEA), the Personnel Constraints Estimation Aid (PCEA), and the Training Constraints Estimation Aid (TCEA) will provide tools for estimating manpower, personnel, and training constraints, respectively. The system performance requirements produced by the SPREA and the MPT constraints produced by the three other aids will be included in Army requirements documents and in system specifications. These documents will provide a comprehensive set of guidelines for prime contractors.

Product 5, the Manpower Determination Aid (MDA), will produce an estimate of manpower and RAM requirements associated with a contractor's design. It will also compare the requirements against the manpower constraints determined by Product 2 and the reliability, availability, and capability requirements produced by the SPREA.

The system performance requirements produced by the SPREA will be also used to determine required personnel characteristic levels in Product 6, the Personnel Requirements Estimation Aid (PREA).

3.3 Steps

The SPREA will consist of an integrated software package. This software will help the analyst in performing the following steps (see Figure 2):

1. Identify a mission that the system must accomplish and the system performance requirements. The Mission Library will contain a list of mission statements that can be copied or modified for this purpose. The system performance requirements represent the minimally acceptable system performance which will result in mission success. These system performance requirements will come from combat models.
   See Section 4.1.

2. Using the SPREA libraries, identify the functions and tasks that comprise the mission. Using the baseline estimates provided by the SPREA, specify the performance criteria (time and accuracy) for each task.
   See Section 4.2.

3. Identify the task sequence by linking the tasks in the proper order. Sample task sequences for a variety of missions will be included in the SPREA libraries.
   See Section 4.3.

4. Identify the tactical and environmental conditions, if applicable.
   See Section 4.4.

5. Run the software aid to model the mission and to calculate the system performance using the performance criteria for the composite functions and tasks.
   See Section 4.5.

6. The SPREA will determine the system performance deficiencies and the "critical paths" that led to them. These deficiencies are the differences between the minimally acceptable system performance (as dictated by the combat models) and the modeled performance (as dictated by the task performance criteria). The software aid will give the analyst the "critical path" through the tasks' times and accuracies that determined the limiting value of each system performance measurement.
   See Section 4.6.

7. With the aid of the SPREA, modify the task performance criteria to correct the deficiencies and re-execute the mission model. Most likely, the SPREA will accomplish this step using backsolving techniques to change the times and accuracies on any tasks where the analyst indicated uncertainty of a performance criterion.
   See Section 4.7.

8. Use the software to generate a report tha details the task performance criteria that were used, the resulting system performance measurements, the reliability, availability, and accuracy estimates for the system, as well as any comments that the analyst wishes to add to the report. This report will also include formatted input to many Army materiel acquisition documents.
   See Section 4.8.

3.4 Automated Components of the SPREA

The automated components of Product 1 reside in the SPREA Applications Manager and consist of data libraries, MPT^2-Specific Templates, Micro SAINT Models, and a SPREA Report Generator. Figure 3 and the following paragraph present brief overviews of each of these components. A more thorough discussion can be found in Section 5 of this concept paper.

3.4.1 Libraries

The SPREA Application Manager will contain four data libraries. The first library, the Mission Library, will consist of system missions and their associated measures of effectiveness. The second library, the Function Library, will contain system functions referenced to the missions that they

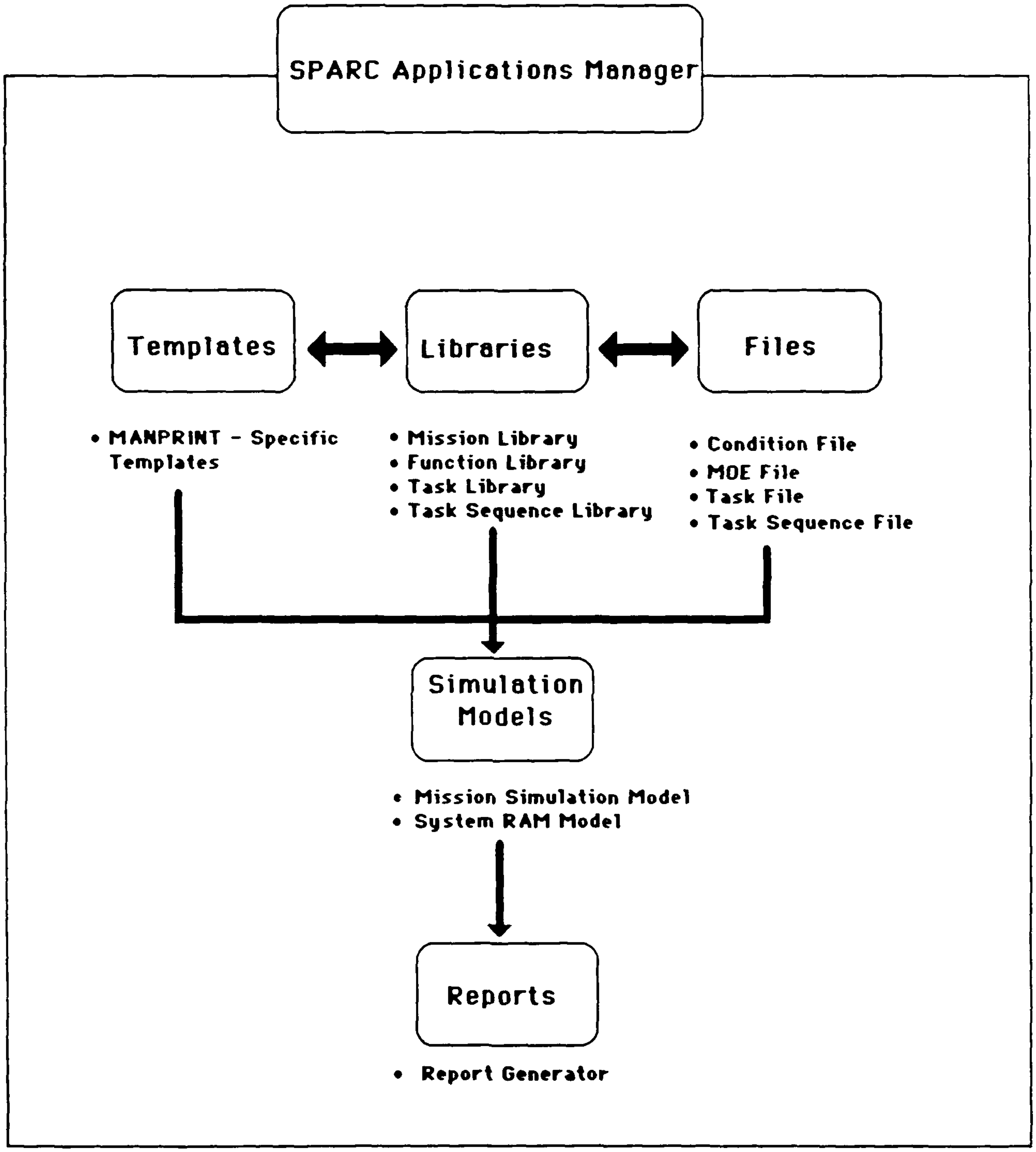


Figure 3. SPREA Applications Manager

are typically associated with. The third library is the Task Library. It will contain system tasks, referenced to particular functions. Finally, since the tasks within the functions will (most likely) be intertwined, the instructions which link the tasks in sequential or parallel fashion will be part of the Task Sequence Library.

The analyst will be able to access the data libraries to pull up missions, functions, tasks, and task sequences that are similar to the ones for which the system is intended. Each of the entries in the libraries will have default information that the analyst will use for "baseline" data. If the analyst modifies a data field that is likely to affect another field, he or she will be prompted to that effect.

The analyst will be able to add, modify, copy, delete, save, and view the elements of the libraries. However, when the analyst is modifying an element of a library, a working copy of the element will actually be changed. This will prevent any loss of data, and ensure that "old" missions will continue to be executable.

If the libraries do not contain entries similar to the ones that the analyst needs, the user interface will allow the analyst to either a) modify an existing entry to create the one that is needed, or b) begin from scratch to enter the needed information. The software will aid the analyst in entering the needed information and store the new task in the library so that it will be available the next time it is required. In this manner, the data libraries will be expanded as new missions, functions, tasks, and task sequences are needed. We recognize that it will be important to safeguard the library data and follow configuration management procedures.

3.4.2 MPT^2-Specific Templates

The MPT^2-Specific Templates will provide the analyst with access to the four data libraries. The user interface will prompt the analyst for information that is necessary to build a simulation model of the system mission. The analyst will use this simulation model to study alternative functional performance requirements.

The templates will aid the analyst in defining new missions, functions, tasks (as well as task performance criteria), and task sequences. They will also enable the analyst to identify the system performance measures of effectiveness (MOEs) and their associated performance requirements. The MPT^2-Specific Templates will prompt the analyst for information about the environmental conditions that are to be modeled, and the tactical situation of interest. It will also allow the analyst to document the mission models.

3.4.3 Models

Micro SAINT will be used as the simulation modeling tool that lies beneath the templates and is transparent to the analyst. Micro SAINT, developed for the Army by Micro Analysis and Design, is a task network modeling tool. Appendix C contains information on Micro SAINT.

There will be two separate Micro SAINT Models in the SPREA: the Mission Simulation Model and the System RAM Model. Both models will be transparent to the analyst (i.e., he or she will never create models with the standard Micro SAINT Model Development tools).

The Mission Simulation Model will be a network of the tasks that the analyst has defined as part of the system mission. In prior steps, the analyst will have defined the links between the tasks, the performance times of the tasks, and the reliability values for each task. Micro SAINT will compile these data, build a simulation model data file, and execute at the analyst's command.

The second Micro SAINT Model will be the System RAM Model. This model will use the system reliability estimate (which is an output of the Mission Simulation Model) and the mission time estimate (also an output of the Mission Simulation Model) to build a simple RAM model. It will also use the analyst's inputs for average system maintenance time, the time between missions, and the number of missions per time unit. The outputs of the System RAM Model will include an availability estimate. This estimate will be the probability that a mission was missed as a result of maintenance.

3.4.4 Report Generator

The SPREA Report Generator will combine the information in the mission simulation, the results of the Mission Simulation Model, and the results of the System RAM Model into a comprehensive report. The Report Generator will ensure that the output information is in a readable and useful format.

3.5 Overview of Approach for Product Development

Product 1 will be developed in two parallel efforts. First, the software that will support the human-computer interface will be developed. This software will be referred to as the SPREA Application Manager. The second product development effort will consist of gathering the data that the analyst will access when he or she is building the mission description. Each of these efforts is described below.

3.5.1 SPREA Application Manager

The SPREA Application Manager will consist of MPT^2-Specific Templates and data libraries. The templates will allow the analyst to define the mission that he or she wishes to model. These components include the functions and tasks that make up the mission, the task performance criteria, the system performance requirements, and the conditions under which the mission will be simulated. This is a crucial element of product development because the software interface will have to be straightforward, logical, and easy to learn.

The data libraries will contain the baseline information that will aid the analyst in building the mission description quickly and easily. The SPREA Application Manager will also include an interface package that will be used to enter the initial data set into the libraries.

3.5.2 Library Data

The second area of product development will consist of gathering the information that will reside in the data libraries. We will provide ARI with initial (although incomplete) libraries of missions, functions, and tasks. At this time, we expect that most of the library entries will be consistent with the Kaplan and Crooks (1980) taxonomy. Table 2 summarizes the data sources we will use to build the libraries and other key elements of the SPREA.

. Table 2. Data Sources of Key SPREA Elements

ELEMENT	DATA SOURCES
Mission Library	Combat Models
	National Training Center Data
	Field Maintenance Data Collection
	Unit Status Reporting System
	Test and Evaluation Data
	ARTEPs
	Requirements Documents
	AMIM
	Army Green Book
	TOE
	Kaplan and Crooks (1980)
	HRTES
	MAAXTAX
Function Library	MAAXTAX
	HARDMAN Comparability Methodology
	ARTEPs
	Kaplan and Crooks (1980)
Task Library	National Training Center Data
	Field Maintenance Data Collection
	Test and Evaluation Data
	ARTEPs
	Requirements Documents
	Kaplan and Crooks (1980)
Task Sequence Library	ARTEPs
	"How To Fight" Manuals
	MAA
	MADP
Conditions File	Kaplan and Crooks (1980)
	TRADOC Scenarios assoc. with MAA
	ARTEPs
	"How To Fight" Manuals
	O&O Plans
	Combat Models
	National Training Center Data
	DT and OT Data

In Task 2 of this effort, we will develop detailed design specifications of the SPREA software components. The final task, Task 3, will involve full-scale product development, including developing the SPREA software components, gathering the data that will reside in the initial data libraries, documenting the product, installing the SPREA, and training the analysts.

SECTION 4 - DETAILED PRODUCT DESCRIPTION

This section presents a discussion of the steps that the analyst will take while utilizing the SPREA. Please refer to Figure 4 for an overview of these steps.

The discussion of each step is organized into the following parts:

1. Input - the data that is required for the completion of this step. Input data consists of two types:

   - External - inputs supplied by sources external to the SPREA
   - Internal - data supplied by sources within the SPREA

2. Process - the process that the analyst will take during this step

3. Output - the data that will be generated as a result of this step

4. User Interface - a brief description of the way that the analyst will be prompted throughout this step

4.1 Step 1 - Identify System Mission and Minimally Acceptable Performance Requirements

This step is separated into two stages. In the first stage, the analyst will identify a specific mission that the system being analyzed must accomplish. During the second stage, the analyst will specify the minimally acceptable performance which the system must attain throughout the mission.

Each of these stages are discussed separately within this section. Section 4.1.1 and its associated subsections discuss mission identification and Section 4.1.2 discusses the system performance specification.

4.1.1 Stage 1 - Identify the Mission

Input. The primary input will be the Mission Library from which the analyst can call up the missions typically associated with the functional area of the system he or she is studying.

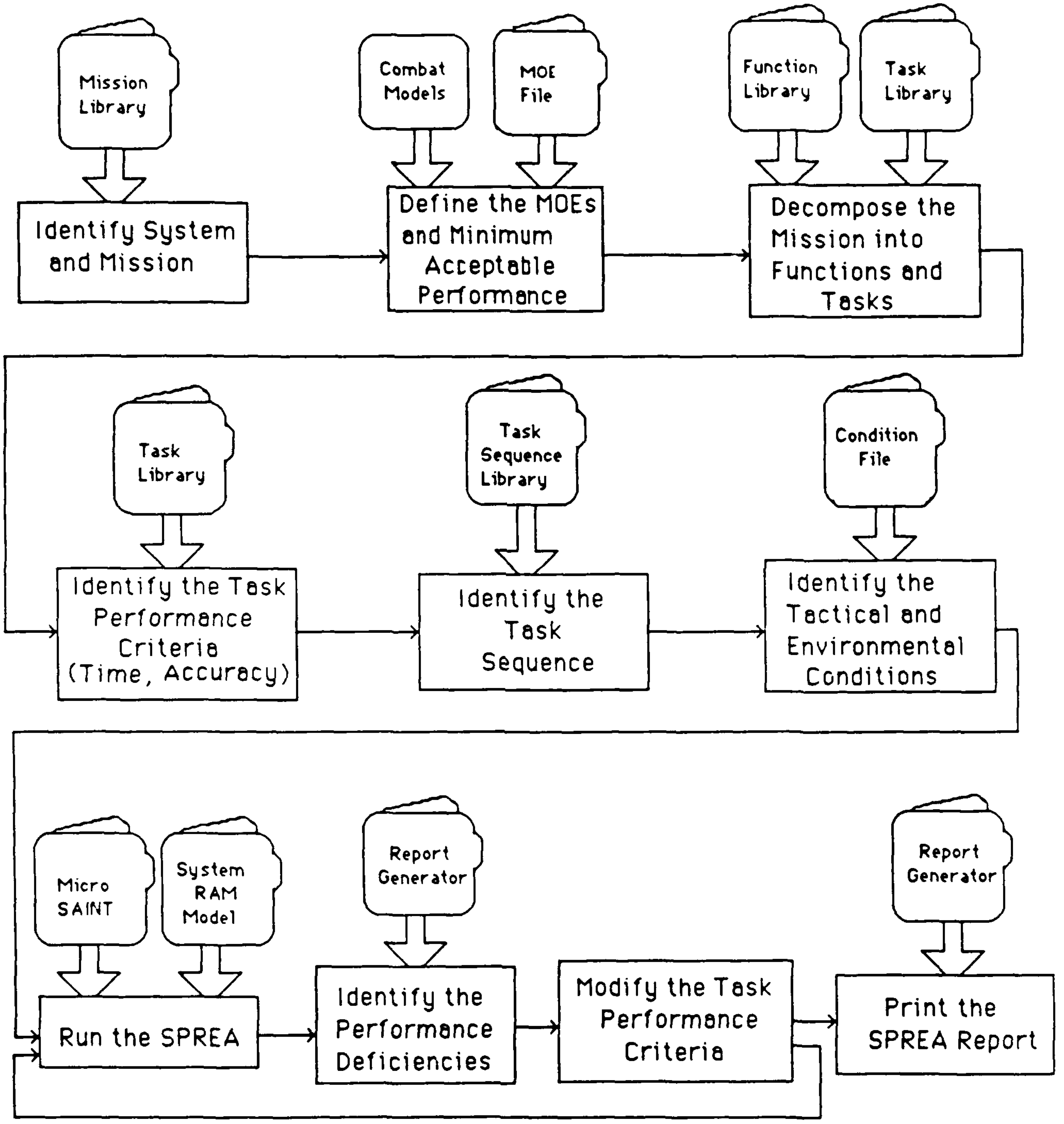

Mission Library
Identify System and Mission
Combat Models
MOE File
Define the MOEs and Minimum Acceptable Performance
Function Library
Task Library
Decompose the Mission into Functions and Tasks
Task Library
Identify the Task Performance Criteria (Time, Accuracy)
Task Sequence Library
Identify the Task Sequence
Condition File
Identify the Tactical and Environmental Conditions
Micro SAINT
System RAM Model
Run the SPREA
Report Generator
Identify the Performance Deficiencies
Modify the Task Performance Criteria
Report Generator
Print the SPREA Report

Figure 4. SPREA Steps

External. The analyst will probably want to modify the mission he or she selects from the library, based on available system documentation and knowledge about the system he or she is analyzing. The analyst will be provided with additional guidance on the construction of acceptable mission statements based on Kaplan and Crooks (1980).

Internal. The internal input consists of the Mission Library. The Mission Library will contain representative mission statements classified within different types of mission areas. We will take these mission statements from mission taxonomies such as Wagner (July, 1986). They will also conform to Kaplan and Crooks (1980), which outlines the essential features of a good mission statement.

Process. The SPREA will aid the analyst in selecting a baseline mission. First, the SPREA will prompt the analyst to classify the system into a mission area. At the present time, we plan to use the following mission areas (from Kaplan and Crooks, 1980):

1. Air defense weapons
2. Armored vehicles
3. Aviation systems
4. Battlefield communication systems
5. C^2 and C^2I systems
6. Combat and tactical support equipment
7. Electronic warfare and surveillance systems
8. Ground transportation equipment
9. Infantry weapons
10. Ordnance systems
11. Target acquisition and designator systems

Each of these mission areas encompasses a number of missions. In fact, some mission areas will contain missions that appear similar, but have different functions. For example, the mission areas "Air Defense Weapons" and "Armored Vehicles" will each have a "Destroy enemy vehicles" mission. However, the functions that a medium range missile will perform to fulfill that mission will be much different than the steps that an M60 Tank will perform.

After the analyst identifies the mission area, he or she will be able to view the associated missions (and the functions that make up that mission) that are contained in the Mission Library. The analyst will then be asked to identify which of the missions most closely fits the one he or she wishes to analyze. If none of the existing missions fits the analyst's needs, he or she can either modify an available mission or build a new mission description from scratch.

As mentioned previously in this document, when an analyst modifies an existing mission, function, or task, he or she will actually be modifying a copy of that existing data item. This is necessary to preserve previous analyses.

If the analyst chooses to build a new mission description from scratch, he or she will be prompted to enter the following information:

1. Mission Area:
2. Mission Name:

Output. The output of this stage will be the identification of the mission area that the system belongs to, as well as the name of the mission that the analyst wishes to simulate.

4.1.2 Stage 2 - Identify Minimally Acceptable System Performance

The SPREA will ask the analyst to enter the minimally acceptable values for overall mission performance. This data is necessary so that 1) the SPREA can help the analyst identify required task performance criteria, and 2) the SPREA can compare these data with the output of the mission model to determine whether the minimally acceptable mission performance was attained in the mission model.

These mission performance data will include the time to perform the mission, mission accuracy, reliability, availability, and maintainability. The analyst will also be prompted to input the system standby time (the time between missions) and the number of missions typically run per time unit.

Input.

External. The analyst will be prompted to enter the minimally acceptable system performance requirements. These data will include mission performance time and accuracy, and system reliability. In order to estimate system availability, the analyst must also enter information on mission standby time, maintenance downtime, and administrative and logistics downtime.

Preferably, the analyst will get these data from combat models. The main disadvantage of this source is that the combat models do not adequately represent the human variable of the system (Van Nostrand, 1986); however, it is the best available source of system performance requirements data at this time. The data may also be available from either the analysis of comparable fielded systems or from the subject matter experts who are familiar with the mission profile. If the analyst does not have the required information on the overall mission requirements, he or she may skip this stage. The SPREA will still predict mission performance using individual task performances, but will not compare it to minimally acceptable mission performance requirements as dictated by the combat models.

Internal. If the analyst has chosen a mission from the Mission Library, then the MOEs and their minimally acceptable performance criteria will be available. The analyst will be able to modify these data to reflect the unique qualities of the system, or he will be able to use the values as "defaults" if it is impossible or difficult to gather more specific data from combat models or comparable systems.

Process. The SPREA will always gather data on a selection of MOEs that will be of interest to all systems. These supplied MOEs are discussed first. There are other MOEs that will vary from system to system and from mission to mission, and will need to be defined by the analyst. These are discussed below in the section on defined MOEs.

Supplied MOEs. System performance MOEs that will always be gathered and reported include mission execution time, mission accuracy, system reliability, system availability, and system maintainability.

A. Mission Execution Time. The amount of time that it took for the system to perform the simulated mission will be gathered automatically and will be included into the SPREA Report.

B. Mission Accuracy. Each task that is included in the mission simulation model will have performance criteria that includes an accuracy estimate (the task performance criteria are discussed in greater detail under Step 2 of this section). We will use these data to calculate an estimate of mission accuracy, which is also a measure of system capability.

C. Reliability. System reliability is another MOE that will be predicted by the SPREA. Reliability is the "probability of a product performing without failure a specified function under given conditions for a specified period of time." Since the SPREA does not have component allocation, we must modify this definition to "Reliability is the probability of a task performing without hardware or software failure for a specified period of time."

The reliability estimate which is reported will be a percentage value. This value will be a result of the accumulation of the individual "probability of task completion" parameters. Please note that this parameter (probability of task completion) has taken into account the execution time of each task. Therefore, by considering task reliability, we waive the need for component allocation to calculate an estimate of system reliability.

The branching and sequencing of the tasks in the mission will be accounted for in the system reliability estimate calculation. This branching will also allow the analyst to specify "catastrophic" failures versus "trivial" failures. The analyst will be able to specify a failure path that will lead to aborting the mission. For an example of the system reliability calculation, refer to Figure 5.

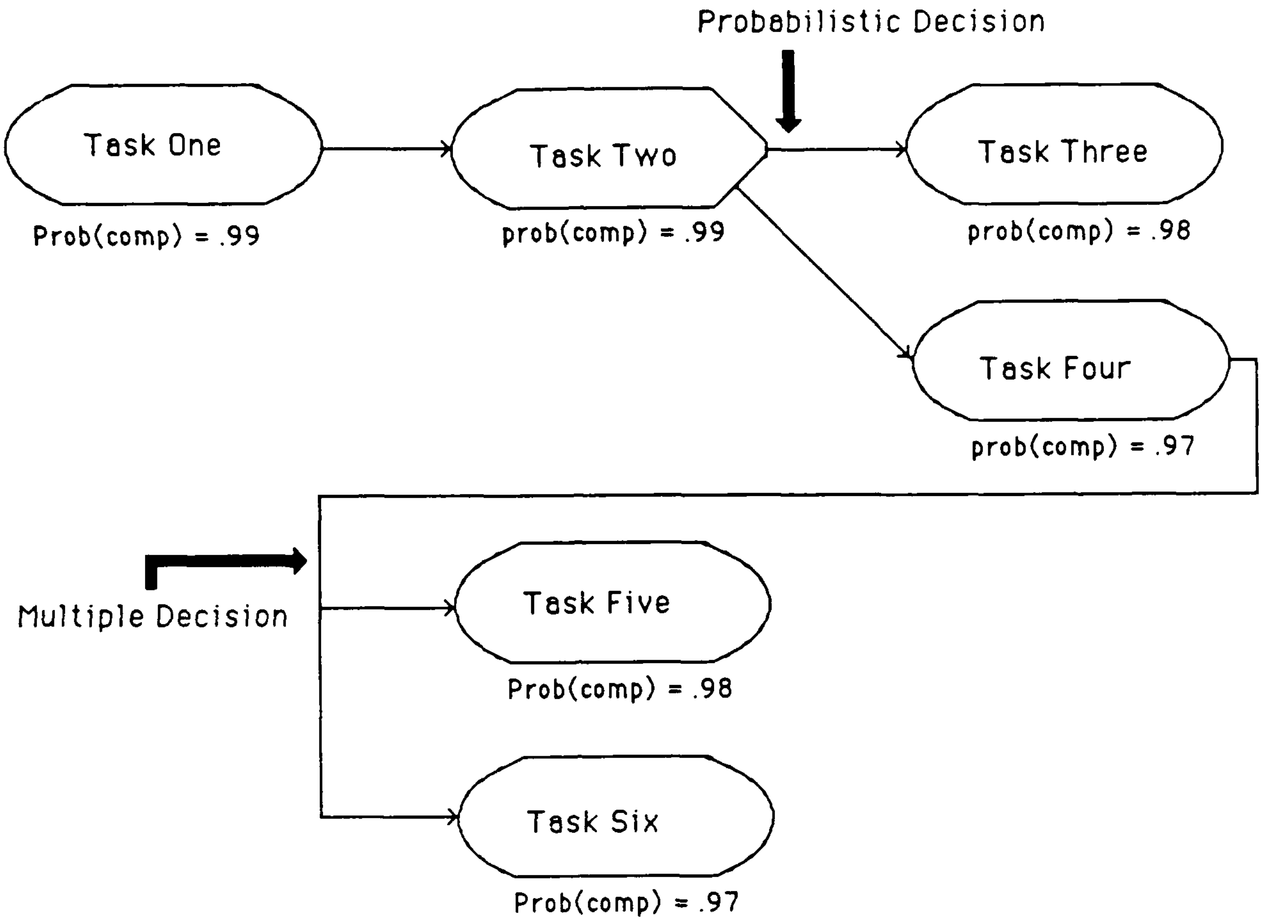
If the success of a particular task is optional (i.e., the success of the mission does not depend upon the success of the task), the analyst will be instructed to let the probability of success criteria for that task default to 100%.

D. Availability. TRADOC DARCOM PAM 70-11 defines Operational Availability (Ao) as follows:

$$Ao = \frac{OT + ST}{OT + ST + TCM + TPM + TALDT}$$

where:

OT = Operating time during a given calendar time period
ST = Standby time (not operating but assumed operable)
TCM = Total corrective maintenance downtime in clock hours during the given time period
TPM = Total preventative maintenance downtime in clockhours during the stated OT period
TALDT = Total administrative and logistics downtime spent waiting for parts, maintenance personnel, or transportation per given calendar time period



Path: Task One , Task Two, Task Four, Tasks Five and Six

Prob(mission completion)= P(comp 1) * P(comp 2) * P(comp 4) * min(P(comp 5), P(comp 6))

= .99 * .99 * .97 * min(.98, .97)

= .92

Figure 5. Reliability Computation

Therefore, an estimate for Ao can be established by:

1. Determining the mission length
2. Determining the standby time (the length of time between missions)
3. Determining the number of missions which will be executed per time unit
4. Determining the average maintenance downtime for the system (TCM + TPM)
5. Determining the average administrative and logistics downtime (TALDT)

This information can be obtained from combat models, from available data on comparable fielded systems, and from subject matter expert data.

The SPREA will build a very simple RAM Model (see Section 5 for a detailed discussion) which will compute the availability estimate. This estimate, as well as the source of the computation, will be documented in the SPREA Report.

E. Maintainability. Since the SPREA does not support component allocation, and since the mission simulation model only portrays one mission at a time, maintainability can not be clearly and meaningfully represented at a task level. Therefore, we will ask the analyst to input an estimate of the system maintainability time (TCM + TPM). This is an estimate of the time that it will take to maintain the system each time that maintenance is required. The analyst will get this data either from subject matter experts who are familiar with comparable fielded systems, or from SDC data from these fielded systems. Since this product does not support component allocation, we feel that the combat models are the most appropriate method of gaining a meaningful maintainability requirement. However, we are open to other ideas on this matter.

This maintenance time estimate will be used in the RAM Model, discussed in Section 5, to provide an availability estimate for the system.

Defined MOEs. The analyst may want to define unique MOEs that are associated with one or more tasks. For instance, target acquisition time and accuracy will almost certainly be collected for missions such as "Destroy enemy vehicles."

If the analyst wants to define new MOEs (or modify existing ones), he or she will be prompted for the following information:

- Beginning task of the MOE
- Final task of the MOE
- Tasks in the sequence between the first and last tasks that must be excluded
- Parameter of interest (e.g., time, accuracy, reliability)
- Expected (estimated) value

If the simulation executes such that the group of tasks which are being measured are repeated, statistics will be kept for each circuit through the set. Final statistics, such as mean and standard deviation of the set of values will be reported in these cases.

The MOE which is predicted by the mission model will be compared to the minimally acceptable performance that the analyst entered. In Step 6, these data will be used to determine the system performance deficiencies. These are the differences between the minimally acceptable performance as reported by the combat models and the performance that was predicted by the mission model. The analyst will be given guidance in Step 7 that will help correct these differences.

Output. The output of this step will be a list of the system performance measures selected for analysis. The analyst will be reminded that each of the system performance requirements will have to be represented with a corresponding MOE.

User Interface

The user interface for this step will have two parts. The first part is the software that the analyst will use to examine the existing entries in the Mission Library. The second part is the software that the analyst will use to modify a mission from the library, to develop a new mission description, or to enter the minimally acceptable mission performance data.

The Mission Library will be in list form, and will be presented via a "spreadsheet" interface. This interface will allow the analyst to page through the available missions, as well as to search for particular character strings. For instance, the analyst can view the library entries for helicopter missions through initiating a search for the mission area "Aviation Systems" (one of the mission areas which will be included in the Mission Library). After examining the missions listed under Aviation Systems, the analyst can select a mission or exit to the menu and begin building a new mission description from scratch.

A "form filling" interface will help the analyst develop new mission descriptions or modify existing ones. The analyst will enter the mission area, mission name, and the values necessary to calculate system RAM criteria. The analyst will use arrow keys or a mouse to travel around the spreadsheet, entering data in the order most convenient.

Since a number of the MOEs are supplied automatically, there will not be a specific user interface developed for the definition of that set of MOEs. The defined MOEs, however, will use a "form filling" interface. This form will include spaces for the analyst to enter the MOE parameters that were discussed in section on defined MOEs. From this form, the analyst will be able to view the task network of the mission he or she is analyzing. This network will help the analyst remember the task names of the beginning, ending, and excluded tasks of the MOE that he or she is defining. It will also remind the analyst of the task sequence.

Both of the interfaces discussed above will provide the analyst with on-line context-specific help and tutorial information about the types of input required for a given task. These features will reduce the amount of training the analyst needs to use the SPREA.

4.2 Step 2 - Identify Functions Tasks, and Their Associated Performance Criteria

Once the analyst has identified the mission for the system to accomplish, he or she must describe the functions and tasks which make up that mission and their performance criteria. The analyst will accomplish this using the information in the Function and Task Libraries.

Input

Function and Task Libraries will be used to limit the amount of information that the analyst must enter.

External. The analyst may want to modify the information from the Function and Task Libraries. If he or she does, depending on their own level of expertise, a range of existing data sources may have to be consulted to obtain information on functions, tasks, and task performance criteria for comparable systems.

Internal. The initial Function and Task Libraries will contain listings of functions and tasks which will be gathered from task taxonomies, such as those included in Wagner (August, 1986) and Wagner (July, 1986). If the analyst is simulating a mission already in the library, a listing of composite functions and tasks will be available.

The Task Libraries will also contain baseline estimates of task performance for each of the existing tasks. These estimates will include time, accuracy, and reliability figures.

We can access a number of data sources to obtain information on system tasks and task taxoncmies for data needed to build the Functicn and Task Libraries. These data sources include a MAA analysis effort, MAAXTAX, HCM, and an Army ARTEPs-based effort.

We will gather task performance criteria data from the National Training Center (NTC) and the ARTEPs efforts, which are supported by the proponent TRADOC schools.

Process

If the analyst chooses a mission from the existing Mission Library, a list of the functions and tasks included in that mission will be available. Each of the functions (e.g., Navigation, Logistics, Information Routing, Prevention of Detection or Location of System) consists of a set of tasks (e.g., Select appropriate maps and navigation aids, Identify present location, Identify destination). Like the missions, the functions and tasks each reside in their own libraries and are accessible to the analyst who wants to add to or modify them. Appendix D contains a preliminary list of the tasks that would be elements of the Task Library.

If the analyst has chosen (in Step 1) to define a new mission, he or she will be asked to list the functions that comprise that mission. The analyst will be able to choose existing functions from the Function Library or, as with the missions, he or she will be able to develop new ones. The analyst will be able to develop new functions either by modifying an existing function or beginning from scratch.

Each function will be decomposed into tasks. The analyst will be able to view the tasks included in the functions he or she has chosen from the Function Library. These tasks will be members of the Task Library. As with the missions and functions, the analyst will be able to access existing tasks, as well as to modify or add tasks.

Each task which resides in the Task Library will have baseline estimates for the following performance criteria:

- most likely performance time

- maximum performance time

  Note: Since we are only interested in minimally acceptable system performance, it will only be necessary to model the mission using the most likely performance time and the maximum possible performance time.

- accuracy of the task

- probability of task completion (reliability estimate). We are using the definition of reliability from Juran (1974) which states "[Reliability is] ... the probability of a product performing without failure a specified function under given conditions for a specified period of time." Therefore, execution time of a task is a factor in the probability of task completion.

The tasks entered in the Task Library will contain baseline estimates of these performance criteria. These baselines will have been derived from performance data on existing systems and will be associated with specific missions, since the performance criteria for a task may depend on the mission itself. Later, the analyst, using the guidance of the SPREA, will be able to modify these baselines to produce a set of values that meets the overall mission performance requirements.

If the analyst does not know what the performance criteria should be and is not sure whether the baseline performance criteria are correct, he or she will be able to instruct the system to assign this value by entering a "?". The SPREA will then use backsolving to assign performance criteria to that task which results in the minimally acceptable system performance. This methodology is discussed in greater detail in Section 4.7 of this concept paper.

The SPREA will be able record how much the performance criteria which was used to meet the minimally acceptable system performance requirements varies from the established baselines that were in the library. The SPREA Final Report will document large deviations. Ultimately, the SPREA will use this information to remind the analyst that the deviations exist, and that they need to be justified (i.e. "technical innovation").

The analyst will be provided with a method of altering the baseline task performance criteria to keep the library data current. This method will be entirely separate from the SPREA to ensure that the baselines are not changed frivolously.

Output

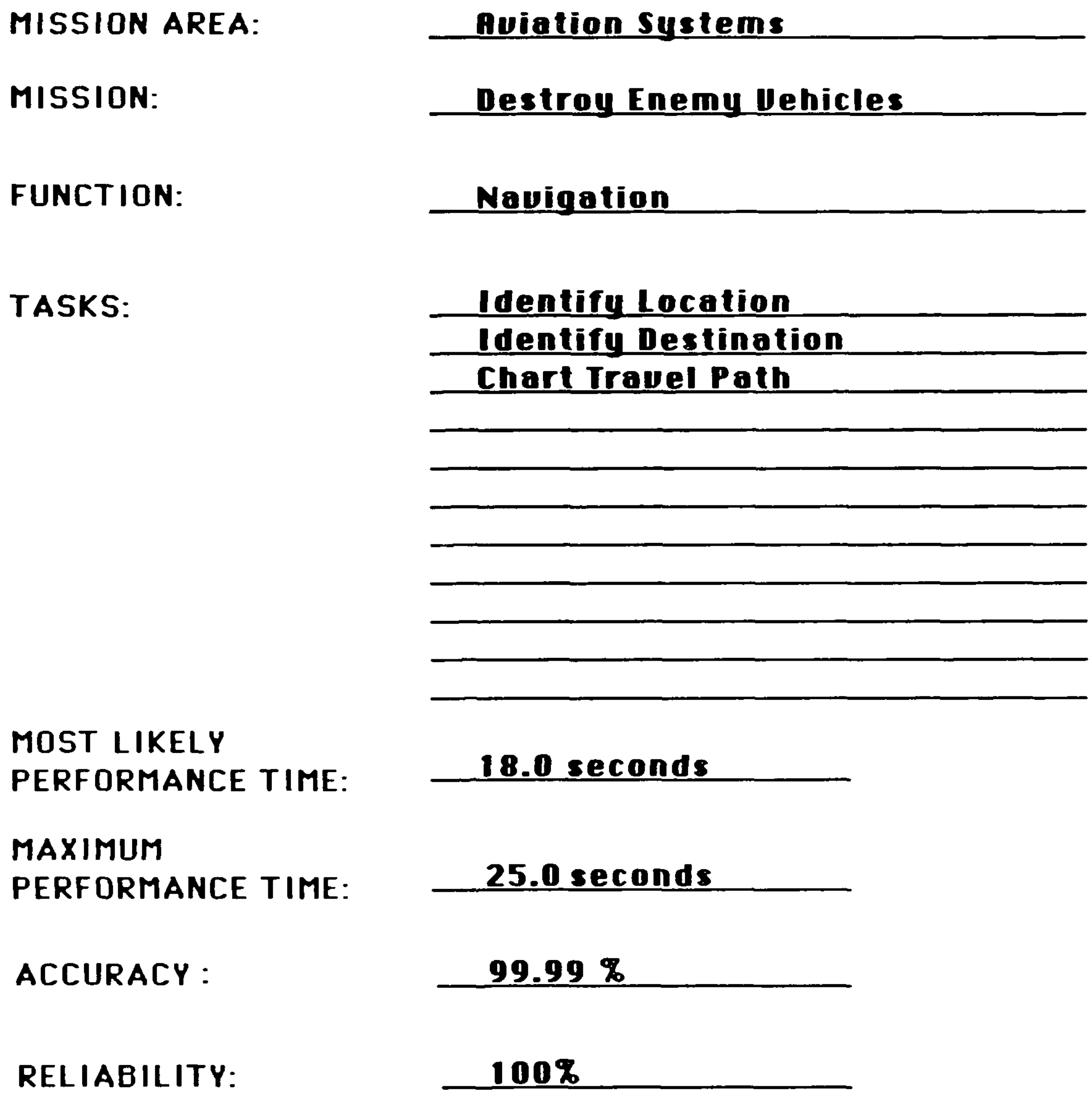
This step's output will be a list of functions and the composite tasks that make up the mission being simulated. The output will also include the tasks' performance criteria.

User Interface

Human Engineering principles state that good user-computer interfaces are consistent between applications. This consistency enables the user to learn the system in less time and to use the system more efficiently. For this reason, the user interfaces described throughout the rest of this section will look much like those described for the first step.

The Function and Task Libraries the analyst will need to access in this step will be presented via a spreadsheet interface. The analyst will be able to page through the spreadsheet, search for a particular character string, or select an entry in the spreadsheet for closer inspection. The Task Library spreadsheet will include fields that contain the performance criteria for each task.

If the analyst selects an entry from a library, the specific parameters (e.g., composite tasks, baseline performance estimates) of that entry will be displayed using a "form filling" interface. An example of such a display is presented in Figure 6. Should the analyst decide to develop function or task descriptions from scratch, a blank form will be available.



MISSION AREA: Aviation Systems

MISSION: Destroy Enemy Vehicles

FUNCTION: Navigation

TASKS: Identify Location
Identify Destination
Chart Travel Path

MOST LIKELY PERFORMANCE TIME: 18.0 seconds

MAXIMUM PERFORMANCE TIME: 25.0 seconds

ACCURACY : 99.99 %

RELIABILITY: 100%

Figure 6. Form Fill-Out Interface Sample

On-line context-specific help will also be available to the analyst at every point while he or she is using this aid.

4.3 Step 3 - Identify Task Sequence

In order to build a mission from the Mission, Function, and Task Libraries, it is necessary to provide links between the tasks. These links will be referred to as the Task Sequence Library. The Task Sequence Library is independent of the Task Library. This enables the analyst to select tasks for a particular mission from the Task Library before dealing with the sequencing of the tasks.

Input

The primary input will be the Task Sequence Library. This library will contain the task sequences of all the missions which are included in the Mission Library. The analyst can use one of the available sequences, or can modify an available sequence to reflect the unique features of the system.

External. The analyst may want to modify the task sequences in the Task Sequence Library. In fact, this modification will be required to do so if any new tasks were added to the mission. Data sources that will be available to assist the analyst in making these modifications are discussed in Section 5.

Internal. The internal input source consists of the existing Task Sequence Library data that have been incorporated into the SPREA and will be available to the analyst. The initial data set that will be used to build this library will come from task analysis data of existing systems.

Process

Task sequence data will be provided for the missions which have been previously incorporated into the SPREA. As with the library data discussed in the preceding steps, these sequence data can be modified or the analyst can define a new task sequence from scratch.

The Task Sequence Library will be very simple. Each task will have a number and the analyst will be asked to identify the first task of the mission and the successor(s) for each task.

Note that accuracy is one of the performance criteria that will have been specified for the tasks in the Task Library. Different successor tasks may apply if the modeled task performance is inaccurate. Therefore, the analyst will have the option of specifying a "failure path" for each task. This failure path gives the analyst a vehicle for specifying tasks whose failures would be catastrophic.

The software will ensure that the analyst has specified links for each task he or she wants to include. The software will also ensure that there are no dead-ends, illogical paths through the tasks, or tasks without paths that lead to them.

The Task Sequence Library supplies an easy method for the analyst to experiment with different task sequences. Since the sequence is independent of the task performance criteria, it will be possible for the analyst to see whether different task sequences will alter the system performance.

Output

The output from this step will consist of a complete task sequence for the mission that is being analyzed. This task sequence will contain the branching directions between the tasks within the mission.

User Interface

The user interface for the Task Sequence Library will also have a spreadsheet format, as described in the User Interface discussion under Step 1. The data in this library will be filed by Mission Area and Mission, so if the analyst has specified a mission which already has a task sequence filed in the library, that sequence will be presented to the analyst automatically. The analyst will also be able to view the other library entries.

If the analyst modifies an existing task sequence, or wishes to build one from scratch, he or she will be presented with a form much like the one in Figure 7. The analyst will be able to enter information in any order.

Mission Area : Aviation Systems

Mission : Destroy Enemy Vehicles

Task	Decision		Following Tasks
Id Position	Success	Single	Id Destination
	Failure	Single	Id Position
Id Destination	Success	Single	Chart Travel Path
	Failure	Single	Id Destination
Chart Travel Path	Success	Multiple	Acquire Target Perform NOE Flight
	Failure	Single	Id Position

Figure 7. Task Sequence Library

4.4 Step 4 - Define Tactical and Environmental Conditions

Knowledge of the tactical and environmental operating conditions of the system is necessary in order to simulate the mission realistically. In this step, the analyst will be prompted for information, and will be advised on how to factor different target characteristics and environmental conditions into the task performance parameters.

Input

External. The conditions under which the new system will perform are the external inputs to this step. As with other inputs that the analyst must supply, if the analyst is not sure of the specific values he or she will be able to try different values and study their effects to determine the model's sensitivity to assumptions regarding different tactical and environmental conditions.

Internal. The internal input consists of a list of the conditions typically associated with different system types performing different missions. The analyst will be able to add comments to these guidelines in order to keep them current.

Process

The SPREA will help the analyst document the specific tactical and environmental conditions which were assumed when the combat models determined the minimally acceptable system performance. The SPREA will prompt the analyst to input the conditions and will include those comments in the final report.

These conditions will include the number of targets and threats, how many targets are to be acquired at once, and the environmental situation.

Output

This step's outputs will include a limited conditions profile that reports: 1) the number of targets and threats and 2) how many can be acquired by the system at once. Also, the output will include the analyst's selection of the conditions under which the mission will be executed.

User Interface

The user interface for this step will consist of a query system. The SPREA will prompt the analyst to enter the operational condition information nd will remind the analyst to consider the conditions in the performance of the related tasks. Finally, the analyst will be asked to include any additional comments in the mission description.

4.5 Step 5 - Run Mission Model and Gather Mission Performance Estimates

In this step, the analyst will tell the SPREA to execute the simulated mission.

Input

External. None.

Internal. The information that the analyst entered in the previous five steps will be used by the SPREA to build a Micro SAINT simulation model of the mission.

Process

This simulated mission will be based on a Micro SAINT model that will be transparent to the analyst. The analyst will be able to control the amount of information which is presented during model execution. If the analyst is interested in the progression of tasks and wishes to view the mission as it executes, he or she will be able to display this information. If, however, he is just executing the model to collect data (in the form of the SPREA Report), the analyst will be able to display less information.

Output

The output of this step will be a completed and executed simulation model of the mission. During execution, the SPREA will collect data of the modeled system performance. The performance data will be presented to the analyst and will also be used in subsequent steps.

User Interface

The user interface for this step will be very simple. The analyst will simply have to enter "GO" from a menu.

4.6 Step 6 - Determine Performance Deficiencies and Their "Critical Paths"

After the mission simulation run has been completed, the SPREA will compile a report. One of the most important items that will be contained in this report is the system performance requirements estimates as calculated in the mission model. These estimates will be compared to the minimally acceptable system performance as derived from the combat models. The differences between the two sets of data are performance deficiencies. This step will also yield the "critical paths" that led to the performance deficiencies (i.e., the tasks and their performance criteria that contributed to the calculation of a system performance value).

Input

External. None.

Internal. The predicted system performance from the mission model, which was a result of the task performance criteria and the minimally acceptable system performance requirements that the analyst entered during Step 1, will be used to calculate the performance deficiencies. The Task Sequence Library will be used to find the critical paths.

Process

The estimates that are reached as a result of the simulation of the mission tasks will be compared with the minimally acceptable system performance requirements that were input by the analyst in Step 1. Any differences between these values will be noted and reported to the analyst.

The SPREA will also report which tasks contributed to the performance deficiencies. This list of tasks will be referred to as the "critical path."

For example, if the analyst were measuring the amount of time required to acquire a target, the critical path would be a list of tasks and their performance times which led to that calculation. As an additional example, if the analyst were measuring the accuracy during a set of firing sequences, the critical path would be a list of tasks and their individual accuracies.

Output

The output of this step will be a list of the system performance MOEs. This list will include the minimally acceptable mission performance that the analyst entered into the system and the mission performance values that were calculated by the SPREA during the simulation run. Any differences between these values will be included. Finally, the critical path which led to each performance deficiency will also be included in the list.

User Interface

As mentioned above, the performance deficiencies and their critical paths will be presented to the analyst in list form. The analyst will be able to print this list, or to use the arrow keys on the keyboard to page through the list.

4.7 Step 7 - Correct Mission Performance Deficiencies

In this step, the analyst will use the SPREA's assistance to change task performance criteria so that the minimally acceptable mission performance requirements are met.

Input

External. Many of these performance deficiencies can be resolved automatically by the SPREA. However, it may be necessary for the analyst to participate in this step if he or she has not directed the SPREA to "play with" specific task performance criteria in order to resolve the differences.

The external inputs which will be necessary for the analyst to resolve the differences between the minimally acceptable system performance and the simulated system performance include the analyst's own expertise and the knowledge of subject matter experts.

Internal. The SPREA will employ backsolving techniques to resolve as many of the deficiencies as possible. There will also be guidelines which will coach the analyst through this step, if it is necessary for him or her to resolve some of the deficiencies. These guidelines will combine a question and answer query system with the data which was compiled in Step 6.

Process

In Step 2 of this process, we explained that if the analyst does not know the correct value for a task performance criterion, he or she will have notified the SPREA that this value is one that the system must assign by using a "?" ("I don't know" response) in the appropriate spreadsheet cell.

The SPREA will have assigned performance criteria to these tasks by defaulting to a best estimate. This best estimate will be the baseline performance value for a task which appears to be a close match for the task in question.

After the simulation executes and the performance deficiencies are calculated, the SPREA will be able to decide which tasks are members of the "critical path" for the modeled system performance criteria which caused the deficiency (see Step 6).

The SPREA will then apply "backsolving" techniques to play with the task performance criteria that were annotated by a "?" during Step 2.

Since the objective of the SPREA is to aid the analyst in determining minimally acceptable criteria, we will inherently have a direction to use in backsolving. When there are system performance deficiencies (i.e., the performance time is too high, accuracy is too low), the aid will set tighter criteria for the tasks on the critical path which will affect those specific deficiencies. These changes can be made in small increments until the deficiency no longer exists.

For example, let's consider a very simple mission model with only three tasks. If, after running the SPREA model, we find that the performance time for the mission is 10, and the minimally acceptable data from the combat model said that it had to be 8.5, then the SPREA will re-examine each task where the analyst input a "?" for performance time. The performance times for those tasks on the mission performance time critical path will be decreased by some small increment. This increment will be calculated as a function of the magnitude of the deficiency and the number of tasks with "?" responses.

The model will be re-executed, and the process will be repeated until the modeled mission performance conforms to the minimally acceptable criteria. The analyst will be presented with the calculated solution and the individual task performance criteria. If the analyst refuses the solution (by seeing that a task performance criteria is set at an impossible level), then the SPREA will start the entire process over, using any new guidelines that the analyst can offer.

If the critical path which led to the performance deficiencies does not contain any task performance criteria that were annotated by a "?", then that means that the performance was calculated using task performance criteria that the analyst explicitly set. In this case, the SPREA will not change these criteria, but will coach the analyst through the process.

The SPREA will offer the analyst a limited amount of guidance if he or she chooses to correct the performance deficiencies. This guidance will probably look something like: "The target acquisition rate is 15% higher than the baseline. Task 15 - Target Identification, performance time is 40% over the baseline value of 10 seconds." Then the analyst will be able to go into the spreadsheet of task performance parameters and edit the performance time for Target Identification if he or she desires.

Output

The output of this step will be revised task performance values for the specific tasks which were a part of the "critical path" which led to the system performance requirement deficiencies calculated during the simulation. The simulation model of the system mission will then be re-executed. This process will iterate until the deficiencies are resolved.

User Interface

In many cases, the deficiencies will be resolved automatically by the SPREA. The analyst will, however, be able to refuse the solutions that the SPREA offers, and will be able to interrupt the

SPREA's search for a solution in order to offer additional information. The analyst will also be able to direct the SPREA to "fix" certain task performance criteria and to "play with" the performance of other tasks.

If the analyst chooses to resolve the deficiencies, he or she will be able to receive a printout of the "critical paths" that led tc the performance deficiencies. The SPREA will give the analyst guidance about the magnitude of the deficiencies. The SPREA will also be able to give the analyst comparison data which describes the differences between the simulated task performance data and the baseline task performance data. This information will aid the analyst in making decisions about which task performance parameters to change in order to meet the system performance requirements.

4.8 Step 8 - Generate Report

The most important output of this product is the SPREA Report which is generated after the simulation model has executed successfully.

Input

External. None.

Internal. The data that has been input by the analyst in the previous steps of this process and the data that is calculated during the mission simulation are inputs into the SPREA Report.

Process

Everything in the SPREA Report will be generated automatically; the analyst will simply have to request the printout.

Output

The output of this step will be the SPREA Report that contains:

- an explicit statement of the missions that were modeled and their composite functions and tasks

- the conditions that the analyst documented
- the required and estimated mission performance parameters, which include:
    - mission execution time
    - mission accuracy
    - a system reliability estimate (i.e., the probability of mission completion)
    - the operational availability requirement
    - a system maintainability estimate
- formatted input to the Letter of Agreement (LOA), the Operations and Organizational (O&O) Plan, the Required Operational Capability (ROC) and the Justification for Major System New Start (JMSNS)

The mission which was modeled, as well as its composite functions and tasks, will be fully documented in the SPREA Report. This documentation will also include a spreadsheet listing of the tasks with their performance criteria. These performance criteria are:

- most likely task performance time
- maximum task performance time
- task accuracy
- probability of task completion (reliability estimate)

Finally, the documentation will supply a network drawing which indicates the predecessor and successor relationships between the tasks. The current version of Micro SAINT (Version 3.0, commercially available in early 1987) already draws these network diagrams, so the development effort for this output is negligible.

The SPREA Report will also document the system performance values for any unique MOEs that the analyst has defined. If, after Step 7, deficiencies still exist between the minimally acceptable system performance and the predicted system performance, then the source of the deficiencies (i.e., the "critical path" through the times and accuracies of the tasks included in each MOE) will be documented, as well as the magnitude of the deficiencies.

In order for the SPREA Report to have optimal utility throughout the Army community, it must be in a format which feeds specific requirements documents. The Directorate of Combat Development typically prepares four documents that will receive input from the SPREA. These documents are the Justification for Major System New Start (JMSNS), the Operations and Organizational (O&O) Plan, the Letter of Agreement (LOA), and the Required Operational Capability (ROC). The format of the SPREA Report will be specifically geared to the formats of these documents. These documents were discussed thoroughly in Section 2.3.1 of this concept paper.

User Interface

The user interface of the SPREA Report will be very straightforward. This interface will be in menu form and will allow the analyst to review and printout all or just portions of the SPREA Report.

SECTION 5 - SOFTWARE OVERVIEW

In this Section, we will define a preliminary architecture for the SPREA. Sections 5.1 through 5.5 will discuss in some detail the software elements that will be part of the SPREA - 1) Application Manager that will include the MPT^2-Specific Templates, , 2) the libraries consisting of the Missions, Functions, Tasks, and Task Sequences, 3) the files that will be created by users through the use of the Templates and the Libraries, and 4) a SPREA Report Generator. Then, Section 5.6 will discuss some of the computer hardware and software issues.

5.1 SPREA Applications Manager

Figure 8 presents the elements of the SPREA Applications Manager. The SPREA Application Manager will consist of MPT^2-Specific Templates which the analyst will use to describe the mission parameters. The templates will be used to access the data libraries, as well as to create new additions to the libraries. They will also be used to enter tactical and environmental scenarios and to evaluate the simulation results.

We will adhere to good principles of human-computer dialogue design in order to produce a human-computer interface that is easy to learn and use. Williges (1986) has published a set of these principles which includes:

- Compatibility Principle - Minimize the amount of information recoding that will be necessary.

- Consistency Principle - Minimize the difference in dialogue both within and across various human-computer interfaces.

- Memory Principle - Minimize the amount of information that the analyst must maintain in short-term memory.

- Structure Principle - Assist the analyst in developing a conceptual representation of the structure of the system so that they can navigate through the interface.

- Feedback Principle - Provide the analyst with feedback and error correction capabilities.

- Workload Principle - Keep the analyst's mental workload within acceptable limits.

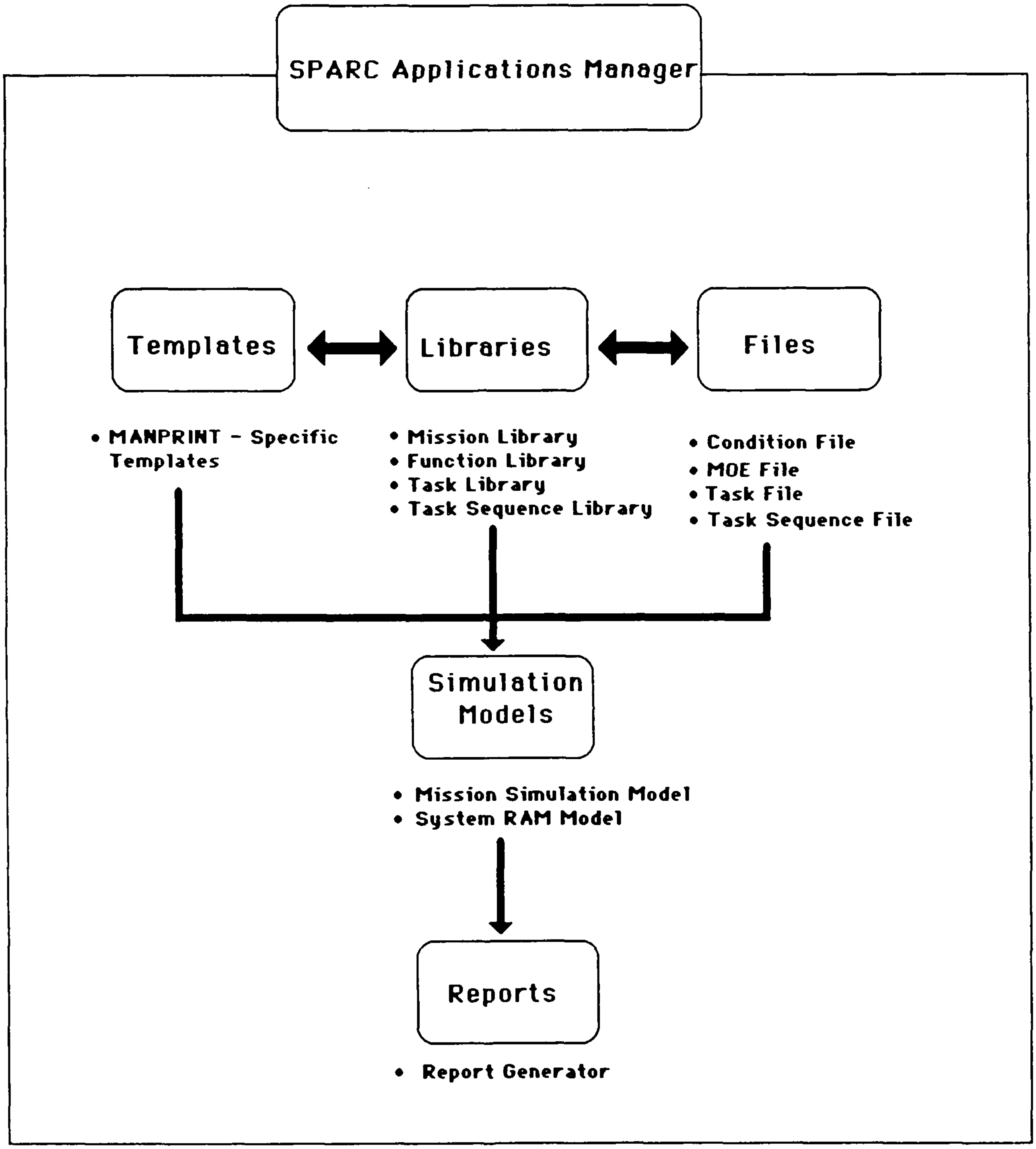


Figure 8. Software Overview

The implementation of software which conforms to the principles listed above will ensure that the analysts will be able to use the SPREA Application Manager with a minimum of training.

The remainder of this section will discuss the SPREA Application Manager in some detail. We are presenting the detail to illustrate the content of the interface, rather than the presentation. The actual software design will be accomplished in Task 2 of this effort.

The first layer of the templates will have a menu-driven format. The characteristics of the decisions which must be made by the analyst are amenable to menu input. A menu interface has been chosen because there are a very limited number of options, and each option has an independent successor decision.

From this menu, the SPREA Applications Manager will transfer control between the templates, libraries, and data files. All of the components of the SPREA Applications Manager are described in the following sections.

5.2 Libraries

There are four data libraries, each of which are discussed in some detail in the next four subsections of this document. Each of these libraries will contain a selected set of data when the SPREA is delivered. This data will be chosen such that it represents systems which are most likely to have MANPRINT considerations.

Mission Library

The Mission Library will contain a list of all the missions that have been entered into the SPREA. These missions will be sorted alphabetically by mission area so that the analyst will be able to locate them easily.

The analyst will be able to view the missions that have been entered. He or she can either choose an existing mission as is, or modify an existing mission (they'll actually be modifying a copy of an existing mission) to meet their needs, or build a new mission from scratch. If the analyst chooses to enter a new mission statement, he or she will receive guidance from Kaplan and Crooks (1980).

In the event that the analyst does enter a new mission statement, the SPREA will prompt for the mission area (for classification purposes); however, the analyst will be able to copy missions from one mission area into another.

The information that is attached to the mission (e.g., composite tasks and sequencing data, threat characteristic data, etc.) will be referenced by mission name and code. Thus, each mission could have a variety of different task sequences or threat possibilities.

The analyst will be prompted for information on mission performance requirements on the three major performance measures -- mission accuracy, system reliability, and system availability. Detailed guidance will be provided to the analyst for obtaining and deriving the minimally acceptable mission performance requirements which are initially input into the SPREA. Ideally, these mission performance requirements should be contained in existing system requirements documents, described in the results of the MAA or MADP which identified the need for the new system. It may be necessary to derive these requirements from combat models. By modeling the capabilities of the current force against the projected threat, the combat models can provide information on projected performance requirements for the new system. This is the best way to set minimally acceptable mission performance requirements since it involves a systematic quantitative comparison of friendly versus threat capabilities.

If the mission performance requirements for the new system are not available or can not be derived from combat models, they can be derived by 1) obtaining performance data from the current system, and 2) increasing or decreasing performance for the existing system by a percentage value to produce what is estimated to be needed to meet the threat. The latter estimate must be made by subject matter experts from the DCD associated with the new system. In order of estimated ·curacy, the most likely data sources for obtaining information on mission performance for the existing systems are:

1. Combat Models - The combat models should have the latest estimates of operational capability for existing systems. These models probably will not produce data which consider the human element of the system (Van Nostrand, 1986); however, they represent the hardware component very successfully.

2. National Training Center - The NTC data base, maintained by the ARI Field Unit at Monterey, contains a wealth of data on the operational capabilities of many Army systems.

3. Field Maintenance Data Collection System - The FMDCS contains extensive data on the reliability, maintainability, and availability of existing weapon systems.

4. Unit Status Reporting System - This is the Army readiness reporting system. It contains estimates of system availability by unit.

5. Test and Evaluation Data - Data from DT or OT testing of the existing system should contain performance estimates for all three parameters.

6. ARTEPs - The ARTEPs for the unit manning the existing system will list the standards of performance which must be achieved on the collective tasks involving the new system.

7. Requirements Documents - Performance requirements should be listed in the requirements documents for existing systems. These requirements may not be stated systematically or may not be at the mission level.

Function Library

The Function Library will be similar to the Mission Library. It will include a listing of all functions which have been entered into the SPREA. The Function Library will be sorted in alphabetic order, which will make it easy for the analyst to locate functions.

When the analyst selects a specific mission, a menu will be displayed that will allow him or her to view the resident functions within that mission. Each of these functions will have been pulled from the Function Library and the list will be sorted into roughly sequential order (taking into account that the tasks from some of the functions may be intertwined). If the analyst wishes to delete or add functions to the ones which are listed for the mission, the SPREA will allow him or her to see a complete list of functions that are in the library.

If the analyst chooses to enter a new function, the SPREA will prompt him or her to ensure that all of the necessary data is entered. One of the required parameters is a list of the function's tasks. If the analyst chooses, he or she will be able to view the Task Library to build the list of previously entered tasks.

The initial function list will be developed using data from existing mission taxonomies. Such taxonomies can be found in Wagner (August 1986) and Wagner (July 1986). In addition, there have been four efforts to develop integrated mission and task taxonomies which we can access for data. These include 1) "Analytical Aids for Conducting Mission Area Analysis," performed under contract to the Army Research Institute, 2) "Mission Area Analysis Experimental Taxonomy (MAAXTAX)," performed in-house at the Army Research Institute, 3) "HARDMAN Comparability Methodology (HCM)," and 4) "Operations Missions Task and Related Core Abilities," which used Army Test and Evaluation Programs (ARTEPs) as a basis for developing functional task hierarchies.

Task Library

We envision that there will be a large number of tasks in the Task Library. Because of this, it will be necessary to classify the tasks so that they can be located quickly. This classification scheme will be centered on "key verbs" such as acquire, aim, assemble, build, etc. The Task Library will be organized alphabetically by key verb.

When an analyst is searching the library for a task, the software must be smart enough to prompt him or her to other key verbs which may describe the task. For example, consider "Recommend main and secondary supply routes" can be replaced by "Identify main and secondary supply routes." To avoid unnecessary replication in the library, we will alert the analyst to potential synonyms.

The initial Task Library will contain an incomplete set of tasks; however, we do intend to supply a complete set of key verbs and synonyms. Currently, key verbs have been taken from the task list in Kaplan and Crooks (1980). This list has less than 100 elements. Assuming that we will limit the set of key verb synonyms to 4 or 5 each and also assuming that many key verbs will not have synonyms, this indexing system should be doable and will be helpful to the analyst.

We envision the input for each task (i.e., the performance parameters) to be well suited for entry into a spreadsheet. This will allow the analyst to quickly enter data in any order. It will also allow the analyst to review the data which has been entered (or was supplied as baseline data). The spreadsheet will allow the analyst to use keypad input and to travel through the document using arrow or paging keys.

A sample task spreadsheet is presented in Figure 9.

Task	Most Likely Perf. Time	Maximum Perf. Time	Accuracy	Probability of Completion	Comments

Figure 9. Sample Task Spreadsheet

The task performance criteria baselines that will be included in the Task Library will be the most difficult data to gather. These data will be gathered from existing systems. In order of estimated accuracy, the most likely data sources for obtaining information on the task performance parameters for the existing systems are:

1. National Training Center
2. Field Maintenance Data Collection System
3. Test and Evaluation Data
4. ARTEPs
5. Requirements Documents

If the analyst wants to generate the own task criteria data, he or she should first try to obtain values from the MAA or MADP results which initiated the need for the system. If the data from this source are not sufficient, the analyst will need to use data from existing systems to estimate task performance. In that case, he or she will need to use the same five sources listed above.

Task Sequence Library

The Task Sequence Library will contain data which control the task sequencing within the missions.

If the analyst is running an existing mission simulation, he or she will be able to access the Task Sequence Library to view the mission's task sequence. The information within the library will be arranged by mission and then by task number. Each task will have designated successor task(s) for two possibilities: task failure or task success. The failure or success of the task will be determined by generating a random number and comparing that to the "probability of success" task parameter value. If the random number has fallen into the "failure" range, then the failure path will be followed. Conversely, if the random number has fallen into the "success" range, then the success path will be followed. In this manner, the analyst can specify "catastrophic" task failures, where the following task on the failure path leads to a mission abort.

When the analyst specifies the following tasks, he or she will have to pecify a decision type. The available decision types will be single task, multiple tasks, probabilistic, or last task. Refer to Figure 10 for an example.

Mission Area : Aviation Systems

Mission : Destroy Enemy Vehicles

Task	Decision		Following Tasks
Id Position	Success	Single	Id Destination
	Failure	Single	Id Position
Id Destination	Success	Single	Chart Travel Path
	Failure	Single	Id Destination
Chart Travel Path	Success	Multiple	Acquire Target Perform NOE Flight
	Failure	Single	Id Position

Figure 10. Task Sequence Library

In order to build the initial Task Sequence Library, operational task sequences will be developed by collecting task sequence information for existing systems. The primary data sources of this information are ARTEPs, "How To Fight" Manuals, and task sequences which may have been developed as part of the MAA or MADP within the functional area. We believe that, unlike task performance criteria, a combat developer with field experience can easily develop task sequences.

If the analyst wants to generate a new task sequence, he or she would use the same sources listed in the previous paragraph to come up with baseline sequences. After learning about the sequences of existing systems, the analyst will then be able to modify the values to reflect the system.

5.3 Files

When the analyst is working on a mission description, the data which he or she is modifying and entering will be stored in files and not in the library itself. This method of storing data will serve to preserve the data which is stored in the libraries while still allowing the analyst to play "what if" with the values of performance parameters, the sequencing of tasks, operating conditions, and the definition of measures of effectiveness. These files are each in the following subsections.

Task and Function File

The functions and tasks to be included in the mission that the analyst is studying will be contained in this file. This file will also contain the performance criteria for each task and function.

The analyst will communicate with this file through the templates discussed earlier. If the analyst chooses to modify or use existing Task or Function Library descriptions, the data in this file will actually be a copy of those descriptions. The analyst will be free to change the file data and to store the data for later use.

A distinction is made between library and file data primarily to ensure that the data in the libraries are only modified or supplemented with "validated system performance data." Ideally, this means that the analyst can enter data in task files and that once the system is fielded (or has passed its acceptance test) the analyst will go back and use the performance test data to update the library. The SPREA will contain an interface that will support this procedure; however, the implementation will have to be left to the analysts themselves.

Task Sequence File

The Task Sequence File is similar to the Task and Function File in that it is the "working copy" of the task sequence of the mission being studied. As with the Task and Function File data, the analyst will be free to play with the task sequences while building the mission description. The new task sequence configuration can be stored as file data until it has been "validated," as discussed in the preceding paragraph. At that point, it can be entered into the library.

Condition File

This file will contain the tactical and environmental or operational conditions under which the system might operate.

The information describing the targets will be very limited. The information will include the number of targets and how many can be acquired at once.

Any other target characteristics, such as how dangerous they are, how difficult they are to kill, how difficult they are to acquire, etc. will be factored into the task criteria of the particular tasks that would be affected by this data. For instance, if a target is particularly difficult to acquire, then the performance time on the "Acquire target" task should be large enough to account for this difficulty. Likewise, if a target is extremely dangerous, this should be reflected in the accuracy criteria of the representative tasks.

The SPREA will offer the analyst some guidance to remind him or her of task performance criteria that might be affected by changing environmental or other operational conditions. The SPREA will prompt the analyst to note which conditions he or she is assuming when setting the task performance criteria, and will record those notes for inclusion into the SPREA Report.

The SPREA will respond to the "Help" command by querying the analyst about performances that might be affected by the condition. This will aid the analyst because the SPREA will approach the problem by systematically examining potential performance effects, and it will be less likely that a parameter will be overlooked.

For example, the SPREA will respond to a help command with queries such as:

"Will this condition affect travel time?" _____
"Will this condition affect reliability?" _____
(If the analyst responds "Yes")
"Will this condition affect reliability of target acquisition?" ____ "Aiming?" ____
etc.

This interchange between the SPREA and the analyst will be recorded for two reasons. First, so that the analyst will be able to obtain a printed copy noting which performance criteria must be changed. Second, so that the system can add this condition to the file.

This query system will provide a vehicle that the analyst can use to keep the Condition File current by adding conditions to the file that were previously unrecognized or by supplementing information that is in the file.

Table 3 presents a preliminary conditions list. It is derived from an earlier list of conditions developed by Kaplan and Crooks (1980). We have eliminated conditions referring to personnel since these elements will now be described as constraints under Products 2 and 3.

To identify the conditions that are most relevant to different types of systems, we will examine the list of conditions included in TRADOC scenarios associated with the mission area. Additional data sources will include ARTEPs, "How To Fight" Manuals, O&O Plans, Combat Models, National Training Center data and DT or OT data.

MOE File

The Measures of Effectiveness File will contain a description of the performance measures that the analyst has collected for a specific mission. This file will include a listing of the names of the performance measures, the initial and final tasks, any tasks excluded from the sequence, and the parameter of interest (e.g., time, accuracy, reliability). The MPT^2-specific templates will be used to build this file, and the file will be referenced by mission name.

Table 3. Preliminary List of Conditions.

I. Environmental

- 1) Weather and Climate (severity and duration)
  - A) Illumination and Visibility
    sunlight - full, marginal, adverse
    moonlight - full, marginal, adverse
    starlight, dusk or dawn, pitch black, artificial lighting flares, direct glare (sun, snow), indirect glare (shadows)
  - B) Temperature
    high, low, normal
  - C) Precipitation
    rain, fog, snow, sleet, sand or dust storm, none
  - D) Wind
    head wind, tail wind, swirling gusts, cross wind, salt spray, wind chill
  - E) Humidity
    high, low, normal
  - F) Atmosphere
    pressure, ozone, lightning

- 2) Terrain
  - A) Ground Slope
    flat or plains, low positive hilly, low negative hilly, high positive mountain, high negative mountain, alps
  - B) Ground Surface
    sandy, rocky, loam, paved, broken paved, broken ground, plowed fields, bare packed, vegetation covered, wooded
  - C) Ground and Water
    light mud, heavy mud, dry, water covered, ice covered, snow covered, subsurface water, moor areas, rivers and streams, lakes, swamps
  - D) Obstacles
    dense vegetation, light vegetation, jungle, hedge row, bodies of water, manmade structures, urban developments, traps, wreckage or debris

Table 3. Preliminary List of Conditions (Continued).

E) Biological
animals, insects, microbiological
pests

3) Induced
A) Type
shock, vibration, acceleration,
nuclear radiation, electronic
countermeasures (ECM),
electromagnetic radiation,
electromagnetic pulse (EMP),
airborne containments, acoustic
noise, thermal energy, modified
ecology, blast, transitional,
chemical

II. Target and Threat
A) Hardware Type
armor, helicopter, aircraft, air defense
systems, missile artillery, C3I, transportation
vehicles, facilities, infantry weapons
B) Unit Type
C) Weapons Type
nuclear, chemical, laser, electronic, directed
energy, conventional
D) Size and Movement
size, stationary or moving
E) Number
single, simultaneous and sequential, noise,
target to non-target ratio
F) Location
minimum range, maximum range, normal range,
azimuth and elevation
G) Speed
maximum, minimum, cruising, rad. alter. of
speed, stationary
H) Concealment
physically, electronically, partially, by
smoke, unconcealed
I) Target Tactics

5.4 Models

There are two separate simulation models which will be developed using the information which the analyst entered into the files discussed in Section 5.3. These models are described below.

System RAM Model

System reliability, availability, and maintainability (RAM) criteria will be modeled in the System RAM Model.

A system reliability estimate will be calculated as discussed in Section 4.5. This calculation will consist of using the task reliability estimates (probability of task completion) to calculate a system reliability estimate (see the discussion in Section 4.5).

In Step 1 of the SPREA process, the analyst will have entered parameters that can be used to estimate system availability and maintainability. These parameters include:

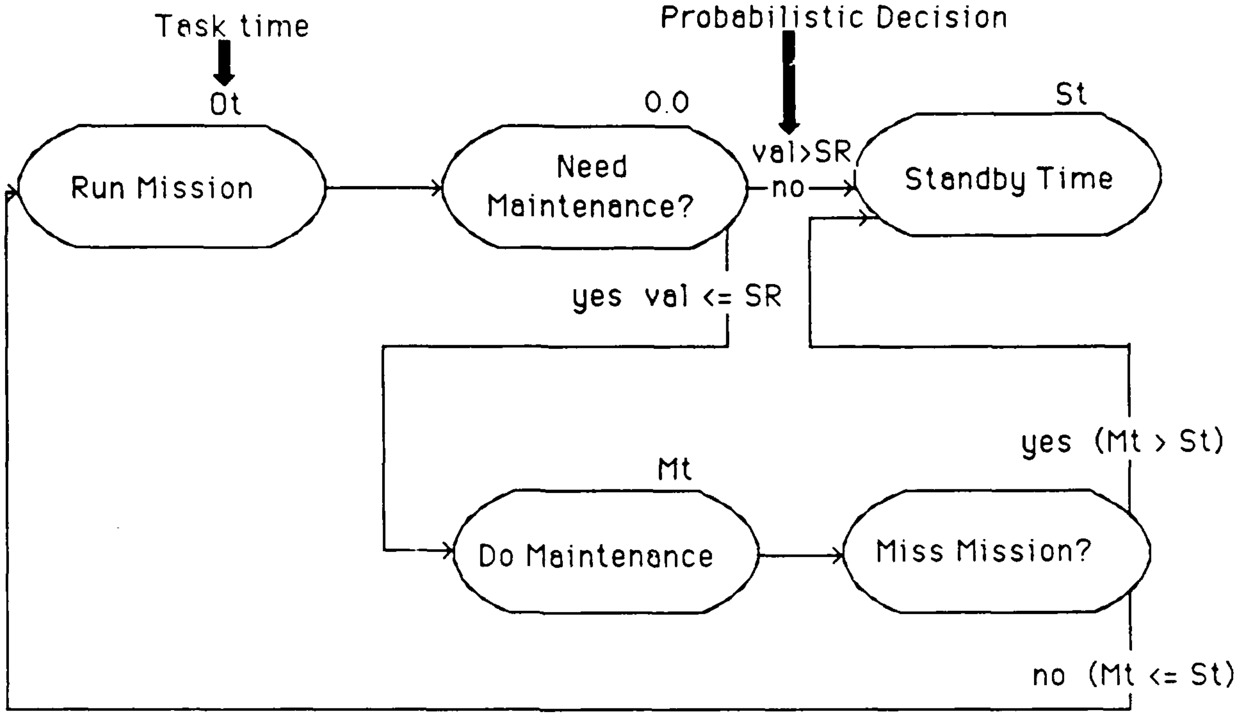
- The number of missions the system is expected to execute per time unit (N)
- The standby time between missions (S_t)
- The expected maintenance time (TPM +TCM)
- Total administrative and logistics downtime (TALDT)
- The last two items can be aggregated to provide an estimate of how long it will take to restore the system to operating order after it fails (M_t).

These values can be used in a very simplistic RAM model. The task network diagram of such a model is presented in Figure 11.

The additional parameters shown in the model (O_t) and (SR) will have been calculated by running the mission simulation model, where:

O_t = Mission operating time
SR = Estimated system reliability

Note that the performance times on each task in the System RAM Model can be changed so that the analyst can play "what if" with the values to determine the appropriate system RAM criteria.



Ot = Mission operating time (calculated in mission simulation)
Mt = Maintenance time (time required to return the syster
operable condition)
St = Standby time (time between missions)
SR = System reliability (calculated in mission simulation)

Figure 11. System RAM Model

From the System RAM Model we can demonstrate that if the time required to maintain the system is greater than the standby time (the time between missions) then the mission has been "missed." Also, a measure of maintainability then, is the probability that the maintenance time is less than or equal to the standby time. A measure of availability can be calculated by stating that "the probability that the system is available is the ratio of how many missions were executed versus the number of missions that were possible."

In algebraic terms:

$$P(\text{available}) = \frac{\#\ \text{missions possible} - \#\ \text{missions missed}}{\#\ \text{missions possible}}$$

and

$$P(\text{maintainable}) = P(M_{ti} <= S_{ti})$$

where

M_{ti} = Maintenance time for mission i
S_{ti} = Standby time (time between missions) for mission i

Data concerning the number of missions possible and the number of missions missed are gathered by running the simulation for the number of times specified by the analyst as the "Expected number of missions per time unit" parameter.

Mission Simulation Model

The Mission Simulation Model will be developed by the SPREA Applications Manager from the data that the analyst entered and subsequently filed in the libraries and working files. This simulation model will be based on Micro SAINT task network simulation, although the model development portion of Micro SAINT will be transparent to the analyst.

Model execution will provide the analyst with a vehicle to watch the simulated mission progress. The analyst will also have tools available which will aid him or her in studying the sequence of the model tasks, and the progression of time as the model executes. These tools will allow the analyst to go back through the MPT^2-Specific Templates and correct any inaccuracies in the model which he or she noticed during execution.

Even though the analyst will not use the Micro SAINT user interface as it presently exists (see Appendix C), it will be necessary to employ the Micro SAINT simulation language for the SPREA Product. Micro SAINT is currently capable of taking data files, compiling any arithmetic expressions and functions, and building a linked discrete simulation model. Micro SAINT is also capable of drawing network diagrams of the model and building timelines of task execution. The interface that the analyst will use to communicate with Micro SAINT will be MPT^2-Specific and will enable the analyst to learn how to use the tool quickly and easily, without confusing the issue with simulation terminology and other extraneous issues.

5.5 Report Generator

The Report Generator software will reformat the data gathered throughout the simulation exercise and present them to the analyst in a usable, readable form.

As stated in Section 3.1, the SPREA will be most useful if it generates a report which feeds directly into specific Army requirements documents. These documents include:

- Justification for Major System New Start (JMSNS)
- Operations and Organizational (O&O) Plan
- Letter of Agreement (LOA)
- Required Operation Capability (ROC)

In addition, the Report Generator will compile the information which the analyst supplied into a readable, complete mission statement.

5.6 Computer Hardware and Software Issues

The SPREA Function Library, the Templates, and other parts of the analyst's tool kit will have to be very comprehensive. They should be able to cover most of the possible systems and missions that an analyst might want to simulate. This could include a large number of possibilities, from different terrain and weather conditions to the varying nature of threats encountered along the way. We would like to support as many scenarios as possible, and not obligate the analyst to construct entirely new models.

Given the scope of these possible missions, it is evident that the SPREA software will have to handle large amounts of data. Furthermore, we would like the system to be "lively"; that is, it should respond to user input in a short amount of time. We envision a two-pronged approach: 1) use a fast processor, and 2) keep the data on-line.

All of Micro Analysis and Design's software was written in the C programming language which is extremely machine independent. Therefore, at ARI's discretion, we may choose to host the SPREA on different machines for different users.

For example, let us consider the microcomputer application. The increase in speed from a PC XT (which uses the Intel 8086 processor) to a PC AT (80286) is impressive. The new machines based on the 80386 are expected to provide even more dramatic improvements in speed. Given our experience in developing and using Micro SAINT, we anticipate that the SPREA could be implemented on a computer based on an 80286-class processor or above. If we begin development on a fast machine from the outset, we should never have to worry about menu response times being too slow.

Micro SAINT runs on an IBM PC, which is able to directly address 640K of memory. This relatively severe memory constraint still allows us to build rather large models. However, for the SPREA software, we feel that 640K of RAM is too constrained. We recommend that the computer on which the SPREA is implemented be able to directly address at least 2 Megabytes of RAM. It should also have at least 20 Megabytes of hard disk storage available. With this amount of storage the software will be able to minimize the time that the analyst spends waiting for disk access operations. In the context of either the VAX or an 80386 based microcomputer, this is a trivial requirement.

In summary, because of the transportability of our software, we propose to develop software for either the VAX or for an 80386 based system with "an eye out" for future applications.

With respect to software, we recommend that all software be written in C. All Micro SAINT data base management and execution software is written in C and we have found it to be extremely powerful and flexible as well as portable.

SECTION 6 - TECHNOLOGY TRANSFER ISSUES

6.1 Training Strategy

Our goal is to design a set of automated (MPT)^2 tools that the user can implement immediately without external training. To accomplish this goal, we will (1) employ a user interface that will allow the system to be used by users who have very little computer experience (see Section 5), (2) provide on-line help to explain alternatives, and (3) automate as many analytical and data collection activities as possible.

In terms of training on how to use the aid, the only external training we believe may be necessary will be a small pamphlet describing the hardware and software needed to run the aid, how to load the aid onto their computer, and what input data they should have on hand before they begin to use aid. We also recommend developing a small manual on how to use the results of the aid in the acquisition process. The experienced MPT user will not need this training. Many times however, a completely inexperienced user who has no background in the acquisition process or in MPT will be assigned to use the aid. This user will need a brief overview of the acquisition process, a brief description of how the aid can help him or her during the acquisition process, and examples of product input and output.

6.2 Means for Achieving Institutionalization

During the development of design specifications for this product (Option 1), we will produce a detailed plan for fielding the product. This fielding plan will describe the distribution of the aid's methods, hardware, software, documentation, training programs (media, instructors, etc.,) to specific Army users in specific Army organizations. The plan will be analogous to the Materiel Fielding Plan developed for Army weapon systems. A draft of this plan will be developed during Option 1, and a final version will be developed during Option 2.

At the present time, we believe that successful implementation will, as a minimum, require the following activities.

Identification of Specific Users

Specific users of each product must be identified and the specific MAP activities and documents into which the product will feed must be described. This will ensure that the product has a use in the "real world." Section 2 describes our approach for accomplishing this.

Incorporation of Users in Product Development

To ensure that the product meets users' needs, users will be included in the product development process. As a minimum, they should use the product during the external demonstration that will take place during Option 2. Ideally, they should also review the final concept papers and the detailed design specifications from Option 1. We stand ready to assist ARI in coordinating user participation in product development.

Incorporation of Acceptability and Usability Requirements into Product Specifications

We have incorporated acceptability and usability requirements into the requirements specification for each aid (see Acceptability and Usability Requirements, Section 2.6.2). These requirements will require that the product include features that will make it easy to use (e.g., clear documentation, on-line help, etc.,). During design, specification (Option 1), we will develop detailed user interface guidelines. To ensure a consistent interface, the SAMC guidelines will be applied to every product.

Instruction of Key Personnel

We propose that "key" personnel receive detailed training at ARI headquarters immediately after ARI has accepted the aid. These key personnel will consist of individuals who can be expected to (1) become experts in using the aid, (2) become instructors in using the aid, and (3) act as consultants for ongoing applications of the aid. At the present time, we recommend that these key personnel consist of selected staff members from ARI's System's Manning Lab., members of ARI field offices who have been designated as MANPRINT support personnel, and members of the MANPRINT policy office within DCSPER.

Demonstrate Aid at User Sites

We also recommend that demonstrations of the aid be provided at all primary user sites. This demonstration could be conducted by contractor personnel or by the key personnel who were trained at ARI headquarters. The demonstration would include hands-on training with the aid software using "real world" examples, describe the benefits of the product, and show how the product can help users produce MAP products.

Software Maintenance

Specific Army organizations must be identified that can continuously update software, documentation, and training to reflect user applications and evolving needs.

Incorporation into Army Training Programs and Regulations

Army training courses for MANPRINT, project management, etc., must be modified to describe how the aid can help users during the MAP. Regulations and pamphlets in these areas must be modified in the same way.

6.3 Estimated Level of Effort Required to Apply the SPREA

We estimate that it will require between 40 and 100 person-hours of effort to apply the SPREA during a major weapon system acquisition process.

REFERENCES

Advisory Group. (1957). Reliability of military electronic equipment report. Office of the Assistant Secretary of Defense (R and D).

Department of the Army. (1985). Product assurance: Army materiel systems reliability, availability, and maintainability (RAM) (AR 702-3). Washington, DC: Headquarters, Department of the Army.

Department of Defense. (1986). Organization and functions mission and major functions of the U.S. Army aviation systems command (AVSCOM Reg 10-1). Headquarters, U.S. Army Aviation Systems Command.

Juran, J. M. (Ed.). (1974). Quality control handbook, third edition. McGraw-Hill Book Company.

Kaplan, J. D., & Crooks, W. H. (1980). A concept for developing human performance specifications. (Technical Memorandum 7-80). (Report No. PTR-2020-80-3).

Parrish, R. N., Gates, J. L., & Munger, S. J. (1981). Design guidelines and criteria for user/operator transactions with battlefield automated systems. Volume III-D: Human factors analysis of user/operator transactions with IISS--FMS--the intelligence information subsystem first milestone. (ARI Research Product No. 81-29). (AD A109 454)

Shaw, B. E., & McCauley, M. E., Ph.D. (1985). Person computer dialogue: A human engineering data base supplement (AFAMRL-TR-85-013). VA: National Technical Information Service.

Smith, S. L. & Mosier, J. N. (1984). Design guidelines for user-system interface software (ESD-TR-84-190) (Prepared for Hanscom AFB). Bedford, MA: The Mitre Corporation.

United States Army. (1980). Materiel acquisition handbook (DARCOM/TRADOC PAM 70-2). Alexandria, VA: U.S. Army Materiel Development and Readiness Command.

United States Army. (1985). TRADOC studies and analysis handbook (TRADOC PAM 11-8). Ft. Monroe: Headquarters, U.S. Army Training & Doctrine Command.

Van Nostrand, S. (September 1986). Model effectiveness as a function of personnel (ME = f(PER)) (Study Report CAA-SR-86-34). U.S. Army Concepts Analysis Agency.

Wagner, M. (July 1986). Human performance capabilities and manpower, personnel, and training inputs in the mission area analysis - preliminary task taxonomy (Report No. E-11606U, Prepared for Mr. David Promisel, USARI/PERI-SM under Contract No. MDA903-86-C-0087).

Wagner, M. (August 1986). Human performance capabilities and manpower, personnel, and training inputs in the mission area analysis-review of literature (Report No. E-11447U, Prepared for Mr. David Promisel, USARI/PERI-SM under Contract No. MDA903-86-C-0087). Andover, MA: Dynamics Research Corporation.

Williges, R. C. (1986). In R. W. Pew & P. Green (Eds.). Human factors engineering short course notes, 27th edition. Ann Arbor, MI: The University of Michigan.

APPENDIX A

SOFTWARE DEVELOPMENT METHODOLOGY

The MPT Decision Support System (DSS) will be developed using the time-proven system process that all successful software organizations employ. This process is outlined in MIL-STD-2167 and DoD Standard 7935. To further ensure success, we have adapted the process described in these standards to meet this project's unique needs. The process, illustrated at Figure A-1, consists of three steps and the products resulting from them.

Step 1 - Requirements Analysis

The requirements analysis identifies the specific functions that the system must perform. High level functional requirements were identified in the concept papers. Detailed functional requirements will be developed in Phase 2, Detailed Design Specifications. The requirements will describe the context, constraints, and functional requirements. Context requirements include:

- The general requirements that the new system intends to meet;
- The environment in which the new system will exist;
- How the new system will interface with other systems;
- How this particular effort fits into any overall long range system development plan; and
- What new technology the effort intends to demonstrate.

The constraints identify:

- Technology limitations;
- Schedule;
- Funding;
- Physical configuration restrictions;
- Political restrictions; and
- The Statement of Work itself.

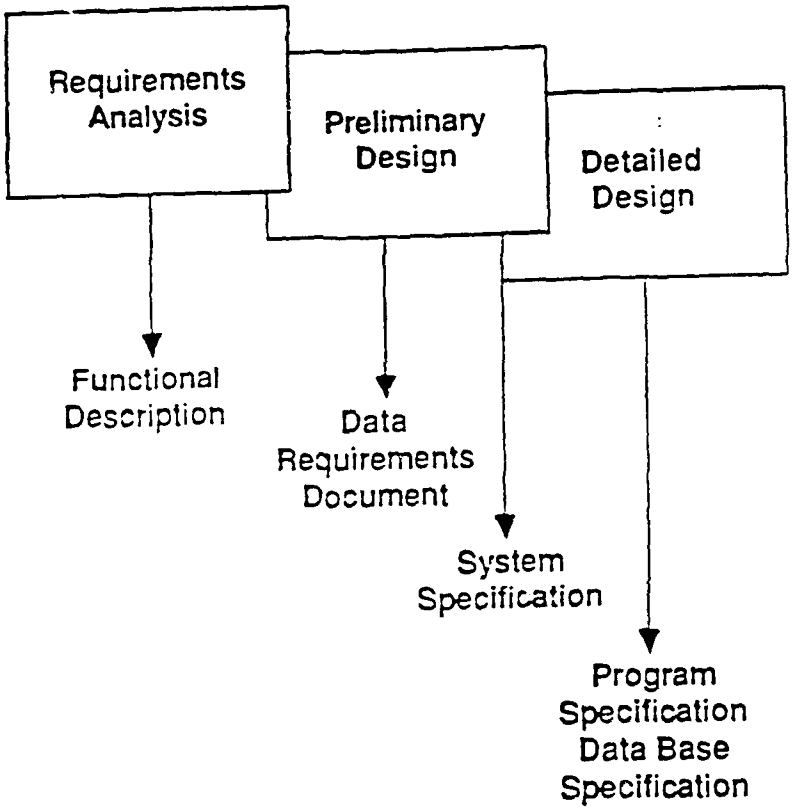


Figure A-1. The System Development Process.

The functional requirements define the functions that will meet the stated goals within the stated constraints, including the input, controls, output, and mechanisms associated with each function.

Step 2 - Preliminary Design

The preliminary design establishes a design concept that meets the needs identified in the requirements analysis. We will conduct this design process early in Phase 2 of the project and submit it for ARI's approval. Once the design concept has been established, we can create a software development plan. This plan will be based upon the level of effort and resources required to implement the design concept. Although a preliminary design concept has already been proposed, the final concept may vary with actual user requirements. The software development plan includes a work plan specifying the tasks that must be conducted and the order in which they must be performed, the computer hardware and software resources needed to develop the system, and a detailed work schedule.

DRC will use an automated program evaluation and review technique (PERT) to create and execute the software development plan. The PERT is an ideal technique for this project since it shows not only when each task is completed, but also how the tasks interrelate. The latter capability is extremely important in this effort because each segment of the DSS depends upon the others.

We will use an IBM PC (or other computer if desired by the COR) to create the software development plan and conduct the PERT analysis. Using a computer in this subtask is critical since the software development plan and the PERT network are complicated and must be updated continually throughout the rest of the project.

The final products of this subtask are a Preliminary Design Technical Report and the Software Development Plan. Although the contract does not require a Preliminary Design Technical Report, it is very important in the system development process. Step 2 consists of the following six activities:

Activity One - Prepare Preliminary Design

The preliminary design effort requires MPT research, system engineering, and ADP experience to generate a system design that will satisf} the user requirements. First, the project team translates the requirements into the output the system needs and develops the logic in arriving at this output. The process then dictates the input information needed to support the process. Next, the team establishes the analytical procedures to support the process. Finally, in order to define the required resources, the team establishes the ADP procedures (i.e., data base management, general processing needs, etc.) The results are documented in a Preliminary Design Technical Report that the COR and the technical members of the development group must approve.

Activity Two - Develop Task Plan

The development of the Task Plan (Software Development Plan) consists of determining the required tasks, the resources needed for these tasks, and a work schedule. In this activity we prepare the Work Plan, which describes the required work and the order in which it must be accomplished. The key to developing a workable plan is having a detailed knowledge of the system development process and extensive experience in applying it to varying systems. The product of this effort is a work flow diagram that not only identifies the individual tasks, but also shows the interrelationships among them. Figure A-2 shows a typical diagram for a single program development effort.

Activity Three - Determine Required Resources

The resources needed to develop the system are determined based upon the products identified in Activity One and the tasks developed in Activity Two. These resources include manpower, computer hardware, computer operations, equipment, facilities, and materials. For a computer development project, this activity also includes analysis of off-the-shelf software resources needs versus newly developed software resources.

Activity Four - Develop Work Schedule

The final activity in determining the task plan is scheduling the work flow and the resources. The schedule must reflect the level of effort for each task, the availability of resources over time, and the interdependency of tasks and resources. There can be trade-offs in the implementation schedules in order to use the resources more efficiently.

Activity Five - Conduct Critical Path Analysis

This activity uses PERT to combine the three activities above into a single critical path analysis. Using a network type of algorithm, the PERT procedure evaluates the interdependency of work packages, the expected time needed to

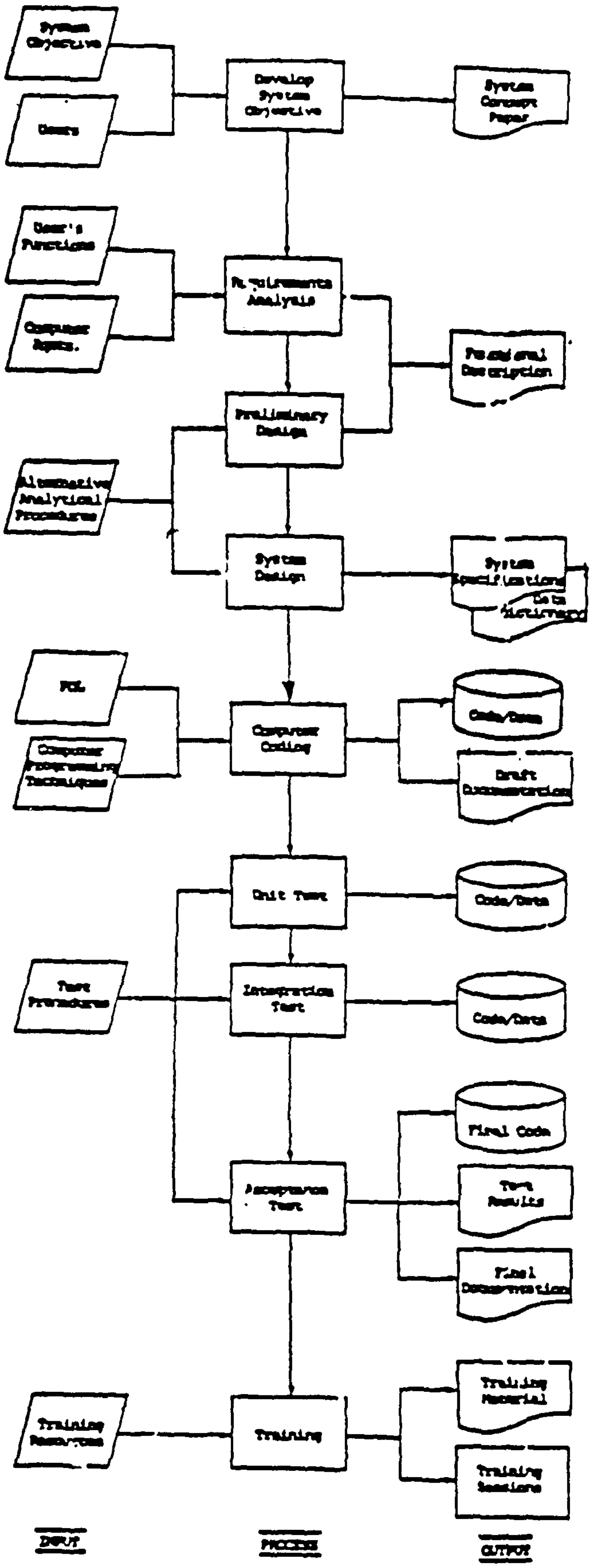


Figure A-2. Sample Work Flow Diagram for a Single Program Development Flow.

complete each task, and the overall effect of varying these times on completing the project. The PERT process is an excellent tool for developing the initial project development program. If properly maintained during the project, PERT allows continual updating of the project status and rescheduling of tasks to meet changing resource allocations and completion dates. We intend to apply the PERT process throughout the development of the Decision Support System.

Activity Six - Prepare Software Development Plan

The final activity in this subtask is to prepare the Software Development Plan. The plan, prepared in accordance with ARI contract report guidelines, contains the information developed in Activities Two through Five and forms the basis of the software development.

Step 3 - Develop Detailed Design

The detailed design transforms the design concept into a highly detailed system design ready for computer application. During this step, the project team determines the specific analytical and data handling procedures. Based upon the approved preliminary design, the project team prepares a detailed system design and determines whether to develop new software or acquire off-the-shelf software. The software is then developed or acquired, unit tested, and integrated with the total system. The team simultaneously determines the data requirements, constructs the data bases, and uses data base management systems. Next, ARI tests the system to ensure that it meets the system-stated requirements before it is accepted. This test includes reviewing and approving the system documentation. Finally, the system is implemented on the host computer, and ARI personnel are trained. We propose to hold this review at the end of Phase 2.

Detailed design includes the following ten activities:

Activity One - Expand Preliminary Design Concept

This activity expands the preliminary design concept developed in Step 2. The expanded version includes a detailed description of the analytical procedures to be employed, the input needed to support the procedures, and the output resulting from the process. The detailed design is presented to the COR at the Critical Design Review (CDR) for approval.

Once approved, the design becomes the Critical Design Baseline. The detailed design specifications are considered the "build-to" specifications and are written at the level where a programmer can write code or adapt an existing software package without further design. During this activity, the team establishes the configuration management program and guidelines for the system documentation.

Activity Two - Analyze Data Requirements

The detailed design conducted on Activity One determines the system's required input and the output forms. Activity 2 is a data requirements analysis that determines the supporting data elements for the input and output. The analysis defines the source of the data, the means of collecting or transferring them to the system, and their management within the system. An important product of this activity is the Data Element Dictionary, which describes each data element by type, source function, and storage requirements.

Activity Three - Locate Existing Software

A key feature of our software development plan is to use existing software whenever possible. This approach drastically reduces the time and risk associated with developing new software, since off-the-shelf software has already been tested and proven effective. In this activity, we search for compatible off-the-shelf software that meets our design specifications. We may use available analytical programs as well as specialized models (Micro SAINT). We also plan to use one of the many data base management systems (DBMSs) on the market to manage the data bases. Hopefully, much of this software can be used as is. If the existing software does not meet our precise needs, we will adapt it rather than attempt to build new software from the ground up. We will build software only if we cannot find any existing programs. If we must develop new software, we will ensure that it is transportable and easily adaptable to similar problems.

Activity Four - Develop Unit Test Code

The most time-consuming software development task is creating or revising computer code and unit testing it. We will develop code using common, standard, high-level programming languages such as FORTRAN and COBOL that are highly reliable and transportable . All code, whether developed by us, adapted from code we have acquired, or taken directly off-the-shelf, must be unit tested to ensure that it meets the requirements Activity One established. The unit test which will use test data generated specifically for the purpose of unit testing, completes the computer development step and establishes the Segment Software Test Baseline.

Activity Five - Conduct Integration Test

The completed software segments are then tested as an integrated package using live data collected during the data base development effort. Once the Quality Assurance (QA) Testing Team has determined that the code meets the system requirements established in Subtask 2.1 and the standards of both the company and the Government, the software becomes the Software System Test Baseline.

Activity Six - Conduct Acceptance Test

While the computer code is being developed and tested, the programming team prepares the system documentation according to the standards identified in Activity One. During the integration testing, the documentation is reviewed for completeness, accuracy, and conformity to the standards. After the QA team approves the documentation (as well as the integrated software), the system goes to the COR for acceptance testing. This time the COR will evaluate the software's accuracy, suitability, and usability using the following criteria:

> Accuracy, or the model's ability to produce accurate results, is measured in terms of technical validity. Technical validity requires that the model's assumptions represent the real world it is intended to simulate, that the data used in the model is correct, that the mathematical formulations are appropriate and correct, and that the errors between the actual outcomes and predicted outcomes do not result from modeling parameters.
>
> Suitability, or the extent to which the model's outputs satisfy the user's requirements, is verified using a requirements traceability matrix. During the requirements analysis, the detailed requirements are listed in a traceability matrix.

As the development continues, the requirements are traced through the preliminary design, the detailed design, the computer code, and the documentation. At the Acceptance Test, the matrix is reviewed to identify each requirement in each development step and its presence in the final software and documentation.

Usability, or the model's usefulness in a realistic environment, is measured by the model's ability to conduct the analyses for which it was developed. This final criterion, typically called "operational validity", assesses the model's ability to produce results acceptable to the user.

Since the model cannot predict the real environment perfectly, operational validity must conclude whether the model's use is appropriate for the observed and expected errors.

Activity Seven - Implement Software and Train Users

The completed and approved system is then implemented on ARI's computer. To ensure its success, we will provide a formal training program for ARI. Built around the system documentation, the training describes the system's operation, preparation of the inputs, and use of the output. The training will include extensive hands-on training at the computer terminal, where most operations occur.

Activity Eight - Integrate Software

The analytical tools developed become part of the integrated (MPT)^2 system, through the integration of their operations and data. This activity establishes the integration between this model and the decision support system tools (especially the Planner and Estimator and the Executor and Interpreter). The sections below describe the required information and the means of integration.

Activity Nine - Evaluate Software

After our contract has ended, the using community will evaluate the software. This activity represents a separate function from the development process. It allows the user to become familiar with the system, while the developer is close at hand to provide assistance and, if desired, revise the model to correct any deficiencies in the original requirements and design. This activity is conducted in parallel with Activity Ten, which provides similar support.

Activity Ten - Provide Operational and Maintenance Support on $(MPT)^2$

After the software is accepted by the user, a maintenance organization must continue to provide operational and maintenance support. This support includes demonstrating how to use the software in an operational environment, helping the user apply the software to actual operational problems, and maintaining the software on the host computer. Any revisions made to the computer code during this time will be reflected in the documentation.

Appendix B

FORMATS FOR ARMY REQUIREMENTS DOCUMENTS OPERATIONAL AND ORGANIZATIONAL PLAN (O&O PLAN)

OPERATIONAL AND ORGANIZATIONAL PLAN (O&O Plan) FORMAT

The O&O Plan describes how a system will be integrated into the force structure, deployed, operated, and supported in peacetime and wartime. The concept establishes required readiness objectives and is the basis for Integrated Logistic Support planning. Initially, the plan should, as a minimum, describe any deficiencies which were identified in the MAA and any constraints applicable to systems development.

I. Purpose - Describe the need for an operational capability to defeat the threat and eliminate an operational deficiency. State where in the MAA the deficiency is identified and how the need was developed from the described deficiency. (The need should be stated in broad characteristics only (e.g., a capability is needed to defeat enemy armor at "x" kilometers)).

II. Threat/ Deficiency - Describe the threat to be countered and the operational deficiency to be eliminated.

*III. Operational Plan - Describe how, what, when, and where the system will be employed on the battlefield and how it will interface with other systems (attach Operational Mode Summary/Mission Profile as an annex). Communications support requirements should be addressed.

*IV. Organizational Plan - Discuss the type units that will employ and support the system and when appropriate, the system(s) to be replaced. (When the system is decided on, include the number of systems estimated to be provided each type unit). This plan will support preparation of the BOIP, the Integrated Logistic Support Plan and identification of key ancillary items.

*V. Personnel Impact - Design of the system should consider personnel skills available to operate and maintain the system. Generation of new MOS should be avoided where possible. (When the system is decided on, include an estimate of the number of people and skills estimated to operate and maintain the equipment, by type unit.) This plan will support preparation of the Tentative Qualitative and Quantitative Personnel Requirements Information (TQQPRI), the Personnel Support Plan, and assist in the LSA process.

Appendix B (Continued)

OPERATIONAL AND ORGANIZATIONAL PLAN (O&O PLAN)

OPERATIONAL AND ORGANIZATIONAL PLAN (O&O Plan) FORMAT
(continued)

*VI. Training Impact - Design of the equipment should consider type and extent of training required. (When the system is decided on, discuss the type and amount of training required and the need for training devices and simulators.) This plan will support preparation of the Training Support Plan.

*VII. Logistics Impact - System must be supportable by the Standard Army Logistics System and use standard tools and TMDE. (When the system is decided on, the proposed levels of maintenance, support concept, Test, Measurement, and Diagnostic Equipment (TMDE), Automatic Test Equipment (ATE), and Built-in Test Equipment (BITE) concepts will be discussed.) This plan will support preparation of the Integrated Logistic Support Plan.

* - Complete information for these paragraphs may not be available when the initial O&O Plan is prepared.

JUSTIFICATION FOR MAJOR SYSTEM NEW START(JMSNS)

JUSTIFICATION FOR MAJOR SYSTEM NEW START (JMSNS) FORMAT

Prepare JMSNS in the format shown below. Do not exceed 3 pages, including annexes. Identify any supporting documentation.

A. Defense Guidance Element. Identify the element of Defense Guidance to which the system responds.

B. Mission and Threat. Identify the mission area (numbers and title) and describe·the role of the system in the mission area. Discuss the DIA-validated projected threat and the shortfalls of existing systems in meeting the threat. Comment on the timing of the need and the general priority of this system relative to others in this mission area. The TRADOC school or Integrating Center must obtain a DIA-validated threat from INSCOM early so as not to delay JMSNS preparation. The classification should be as low as possible; NOFORN data should not be included. DIA threat documentation should be referenced in lieu of higher classification. If the need is not threat driven describe the basis for the need (e.g., cost savings).

C. Alternative Concepts. Describe the alternatives which will be considered (including product improvements) and, when appropriate, the alternative selected, the reasons for rejecting those that have not been selected, and any further tradeoffs that remain for the selected system.

D. Technology Involved. Discuss maturity of the technology planned for the selected system design and manufacturing processes, when appropriate, with particular emphasis on remaining areas of risk.

E. Funding Implications. Provide gross estimates of total RDT&E cost, total procurement cost, unit cost and life-cycle cost. Discuss affordability. See Appendix D, this Handbook, for funding format.

F. Constraints. Describe, as applicable, key boundary conditions for satisfying the need, such as survivability; logistics, manpower and personnel constraints in both quantity and quality; standardization or interoperability within NATO or other DOD Components; and critical materials and industrial base required.

G. Acquisttion Strategy. Provide summary of salient elements of proposed acquisition strategy -- program structure, competition, contracting, etc.

LETTER OF AGREEMENT (LOA)

LETTER OF AGREEMENT (LOA) FORMAT

The Letter of Agreement (LOA) will be in the format below. Limit information to that necessary for a HQDA decision. The basic document should not exceed four pages. In the LOA, use less detail and broader performance bands than in the ROC, JSOR, LR, and TDR. Terms in each paragraph of the LOA will evolve into more specific terms in the ROC, LR and TDR. Include in the LOA all alternative system concepts recommended for demonstration and validation.

1. TITLE

a. Give a descriptive title for the program.

b. CARDS reference number.

2. NEED/THREAT. State what is needed. Briefly describe the threat and operational/training deficiency need for the system. Include the enemy's capability to detect, identify, locate, avoid, suppress, destroy, or otherwise counter the system. Describe the responsive threat over time to support evolutionary development when applicable.

3. TIMEFRAME AND IOC. State the timeframe in which the new or improved system is needed.

4. OPERATIONAL & ORGANIZATIONAL PLAN: In a brief paragraph state --

a. How the equipment will be used;

b. Geographical areas of use;

c. Weather and climatological factors to be considered during equipment operations;

d. Battlefield conditions (such as ECM, smoke, and dust) in which the system will operate; and

e. The type of units that will use and support the equipment.

Attach the mission profile to the LOA as an Annex.

5. ESSENTIAL CHARACTERISTICS. Describe only main operational features of the system. Included are counter-countermeasure capabilities, health, physical security, safety and human factors engineering requirements, and reliability, availability, and maintainability (RAM) requirements. Performance must be responsive to battlefield environmental conditions of continuous combat (such as full ECM, smoke, aerosols, rain, fog, haze, and dust).

LETTER OF AGREEMENT (LOA)

LETTER OF AGREEMENT (LOA) FORMAT
(continued)

Express performance and reliability characteristics in bands of performance. Those which are not suitable for banding will be stated as single values. During development, commercial, other service, NATO, or other allied nation characteristics of existing or programed systems should be considered for inclusion. This will be with a view toward establishing a basis for interoperability, co-production, or standardization. Bands of performance should be flexible enough to consider competing systems of other services or allied nations. Stated bands of performance, or single value characteristics will be adjusted only after the combat and materiel developers agree that such changes are necessary. DCSOPS will approve changes for documents previously approved by DCSOPS. The requirements and provisions for the following must be considered.

a. Interoperability;

b. Continuity of operations (CONOPS);

c. Security;

d. Reliability, availability, and maintainability (RAM) derived from mission performance parameters.

e. Standardization, including commonalty for hardware and software to which the system will adhere;

f. Nonnuclear/nuclear survivability; NBC contamination/decontamination survivability;

g. Individual/collective protection equipment;

h. Adverse weather and reduced visibility conditions (smoke and obscurants) operations, and military operations on urbanized terrain (MOUT) where applicable;

i. Communications;

j. Operation transportability, such as: transportable in C-141 type aircraft requiring not more than....hours teardown and....hours setup by operator and crew, etc.

6. TECHNICAL ASSESSMENT. In the LOA, divide this paragraph into operational, technical, logistics, training, and manpower subparagraphs. In each, describe what the combat and materiel developers, logistician, trainer, and personnel administrator must do to produce the total system. Include a listing of major events and dates.

LETTER OF AGREEMENT (LOA)

LETTER AGREEMENT OF AGREEMENT (LOA) FORMAT (continued)

7. LOGISTICS SUPPORT PLAN. Briefly describe the logistics support plan. The logistics support plan will be available for evaluation during OT I.

8. TRAINING ASSESSMENT. Discuss the need for system training devices. When required include description as an annex. (See p. 6.20 for format.) New Equipment Training (NET), operator and maintenance personnel training, and technical manuals and training material requirements will be stated in terms of needs for both the institution and unit training levels. The training support plan will be available for evaluation during OT I.

9. MANPOWER/FORCE STRUCTURE ASSESSMENT. Estimate manpower requirements per system, using unit, and total Army by component (Active, ARNG, USAR). Identify manpower savings resulting from replaced systems, if any. Include a statement to require an assessment of alternatives to reduce manpower requirements and an assessment of force structure implications resulting from system inclusion in the total force by component. If the force structure assessment exceeds current programed force structure levels then identification of force structure tradeoffs within mission area or mission elements are required. Tradeoffs analyses are addressed to the degree necessary to bring the force structure assessment within current programing levels, if possible. The personnel support plan will be available for evaluation during OT I.

10. RATIONALIZATION, STANDARDIZATION, INTEROPERABILITY. Discuss other Services, NATO, and other foreign interest in the program. Identify similar programs contemplated by other services, NATO or other allies.

11. LIFE CYCLE COST ASSESSMENT. See appendix 1.

12. MILESTONE SCHEDULE. A listing of significant events with dates to occur between approval of the LOA and next scheduled milestone review. The following should be included: LOA approval, OT/OT/other test (Market/User Survey for OTS), and next scheduled milestone review.

APPENDIX 1 - Life Cycle Cost Assessment - Provide life-cycle costs using mainly summary parametric estimating techniques. State the major life-cycle phases of R&D, investment, and operation and support. Also include the design to cost goals. As much as possible, show the estimated cost of major items or components below the system level. These data should be consistent with the Materiel System Requirements Specification (MSRS) and Baseline Cost Estimate (BCE).

Appendix B (Continued)

LETTER OF AGREEMENT (LOA)

LETTER OF AGREEMENT (LOA) FORMAT
(continued)

ANNEX A - Coordination. List all major commands, other Services, allied nations, and activities with whom the LOA was coordinated. Provide full rationale for nonacceptance of comments, if any.

ANNEX B - Operational Mode Summary/Mission Profile Annex. List tasks and conditions for frequency and urgency viewed for system employment in military operations. The mission profile is logically derived from the O&O Plan. It provides the starting point for developing the system characteristics. See p. 5.23 for format for mission profile.

ANNEX C - COEA Annex. Executive summary of the COEA. Classify as required. Withdraw after HQ TRADOC approval of the LOA and handle as a separate document for transmittal as needed.

ANNEX D - Rationale Annex. Support various characteristics stated in the LOA. This provides an audit trail and rationale for determining how the characteristics were derived.

ANNEX E - RAM Rationale Annex. Executive summary of the RAM Rationale Report. Support the stated RAM characteristics with a logical argument that begins with the task frequency, conditions and standards described and analyzed in the MAA. This provides an audit trail and rationale for determining how the characteristics were derived. TRADOC/DARCOM Pamphlet 70-11 contains guidance on the preparation of both the RAM Rationale Report and the RAM Rationale Annex.

ANNEX F - Training Devices. When required, include description of needed training devices in format on p. 6.20. A separate annex is required for each training device.

NOTES:

1. All annexes will accompany the LOA until it has completed TRADOC and DARCOM staffing.

2. Send A, B, and F with the LOA when forwarded to HQDA for approval.

REQUIRED OPERATIONAL CAPABILITY (ROC)

REQUIRED OPERATIONAL CAPABILITY (ROC) FORMAT

The Required Operational Capability (ROC) is in the format below. Limit information to that necessary for a HQDA decision. The basic document should not exceed four pages.

1. TITLE

   a. Give a descriptive title for the program.
   b. CARDS reference number.

2. NEED/THREAT. Briefly describe the operational/training deficiency need for the system and the reactive threat to the system. Include the enemy's capability to detect, identify, locate, avoid, suppress, destroy, or otherwise counter the system. Describe the responsive threat over time to support evolutionary development when applicable.

3. TIMEFRAME AND IOC. State the IOC date including IOCs for successive evolutionary models, when appropriate.

4. OPERATIONAL AND ORGANIZATIONAL PLAN (O&O Plan). In a brief paragraph state:

   a. How the equipment will be used;
   b. Geographical areas of use;
   c. Weather and climatological factors to be considered during equipment operations;
   d. Battlefield conditions (such as ECM, smoke, and dust) in which the system will operate; and
   e. The type of units that will use and support the equipment.

5. ESSENTIAL CHARACTERISTICS. Describe only main operational features of the system. Included are counter-countermeasure capabilities, health, safety and human factors engineering requirements, and reliability, availability, and maintainability (RAM). Performance must be responsive to battlefield environmental conditions of continuous combat (such as full ECM, smoke, aerosols, rain, fog, haze, and dust).

Express performance and reliability characteristics in bands of performance. Those which are not suitable for banding will be stated as single values. During development, commercial, other Service, NATO, or other allied nation characteristics of existing or programed systems should be considered for inclusion with a view toward establishing a basis for interoperability, co-production, or standardization. Bands of performance should be flexible enough to consider competing systems of other Services or allied nations. Stated bands of performance, or single value characteristics are adjusted only after the combat and

REQUIRED OPERATIONAL CAPABILITY (ROC) FORMAT
(continued)

materiel developers agree that changes are necessary. DCSOPS will approve changes for documents previously approved by DCSOPS. The requirements and provisions for the following must be considered:

a. Interoperability;
b. Continuity of Operations (CONOPS);
c. Security;
d. Reliability, availability, and maintainability (RAM) derived from mission performance parameters;
e. Standardization, including commonality for hardware and software to which the system will adhere;
f. Nuclear survivability; NBC contamination survivability;
g. Individual/collective protection equipment;
h. Adverse weather and reduced visibility (smoke and obscurants) operations, and military operations on urbanized terrain (MOUT) where applicable;
i. Communications.
j. Operation transportability requirements, such as: transportable in C-141 type aircraft requiring not more than....hours teardown and....hours set by operator and crew; etc.
k. P3I

6. TECHNICAL ASSESSMENT. In the ROC, include a brief paragraph about the technical effort required. Address major areas for full scale development in terms of scope, technical approach, and associated risks in high, medium, low, or similar categories. For NDI items, briefly outline completed or planned market survey efforts and/or military suitability evaluations.

7. LOGISTICS SUPPORT PLAN. Briefly describe the logistics support concept. The logistics support package will be tested during OT II.

8. TRAINING ASSESSMENT. Discuss the need for system training devices. When required, include description as an annex to the ROC. (See p. 6.16 for format.) New equipment training (NET) operator and maintenance personnel training, and technical manuals and training materiel requirements will be stated in terms of needs for both institution and unit training levels. The training support package will be tested during OT II.

9. MANPOWER/FORCE STRUCTURE ASSESSMENT. Estimate manpower requirements per system, using unit, and total Army by component (Active, ARNG, USAR). Identify manpower savings resulting from replaced systems, if any. Include a statement to require an assessment of alternatives to reduce manpower requirements and an assessment of force structure implications resulting from system inclusion in the total force by component.

REQUIRED OPERATIONAL CAPABILITY (ROC)

REQUIRED OPERATIONAL CAPABILITY (ROC) FORMAT
(continued)

If the force structure assessment exceeds current programed force structure levels then identification of force structure tradeoffs within mission area or mission elements is required. Tradeoffs analysis are addressed to the degree necessary to bring the force structure assessment within current programing levels, if possible. The personnel support package will be tested during OT II.

10. STANDARDIZATION, INTEROPERABILITY. Discuss other Service, NATO, and other foreign interest in the program. Identify similar programs contemplated by other Services, NATO or other allies.

11. LIFE CYCLE COST ASSESSMENT. See appendix 1.

12. MILESTONE SCHEDULE. A listing of significant events with dates to occur between approval of the ROC and next scheduled milestone review. The following should be included: ROC approval, DT/OT/other test (Market/User Survey for OTS), and next scheduled milestone review.

APPENDIX 1 - Life-cycle Cost Assessment. Provide life-cycle costs using mainly summary parametric estimating techniques. State the major life cycle phases of R&D, investment, and operation and support. Also include the design-to-cost goals. As much as possible, show the estimated cost of major items or components below the system level. (These data should be consistent with the Materiel System Requirements Specification (MSRS) and Baseline Cost Estimate (BCE). (See app D, p. D.7, this handbook, for format).

ANNEX A - Coordination. List all major commands, other Services, allied nations and activities with whom the ROC was coordinated. Provide full rationale for nonacceptance of comments, if any.

ANNEX B - Operational Mode Summary/Mission Profile Annex. List tasks and conditions for frequency and urgency viewed for system employment in military operations. The mission profile is logically derived from the operational/training concept. It provides the starting point for developing the system characteristics.

ANNEX C - COEA Annex. Executive summary of the COEA. Classify as required. Withdraw after HQ TRADOC approval of the ROC and handle as a separate document for transmittal as needed.

ANNEX D - Rationale Annex. Support various characteristics stated in the ROC. This provides an audit trail and rationale for determining how the characteristics were derived.

REQUIRED OPERATIONAL CAPABILITY (ROC)

REQUIRED OPERATIONAL CAPABILITY (ROC) FORMAT (continued)

ANNEX E - RAM Rationale Annex. Executive summary of the RAM Rationale Report. Support the stated RAM characteristics with a logical argument that begins with the task frequency, conditions, and standards described and analyzed in the Mission Area Analysis (MAA). This provides an audit trail and rationale for determining how the characteristics were derived. TRADOC/DARCOM Pamphlet 70-11 contains guidance on the preparation of both the RAM Rationale Report and the RAM Rationale Annex.

ANNEX F - TRAINING DEVICE ANNEX. Include when appropriate. (See p. 6.20 for format.) A separate annex is required for each training device.

NOTES: 1. Send annex A with each requirements document.

2. Annex F (when prepared) must accompany the ROC to HQDA for approval as a package.

3. Send the TBOIP/TQQPRI with the ROC to HQDA for approval. When the TBOIP/TQQPRI are not submitted, the transmittal letter will contain a statement about the projected submission date.

APPENDIX C

PRELIMINARY TASK LIST

Acquire
- obstacle to be dealt with
- target; include the judgment of distance to target

Activate
- hardware protective device(s)

Adjust
- aim, following miss
- fire of attacking unit(s)
- launch based on location of detonation in relation to target

Aim
- weapon system. This involves a procedure which results in the system being adjusted for the azimuth and elevation of the target
- mine
- grenade

Apply
- anti-jamming procedures
- transmission security procedures

Arm
- mine
- system

Assemble
- communications device(s)
- system

Assign
- weather indicator collection tasks
- intelligence collection tasks to maximize receipt of indicators according to their priorities
- security classification and method for maintaining that classification
- confidence levels to the projection(s)
- probabilities to weather projections

Assume
- protective position for crew and passengers

Attach
- cables to anchors or winches
- to appropriate part(s) of person, harness, etc.

Authenticate
- transmissions

Calibrate
- system including boresighting and collimating
- system components

Camouflage
- system (System camouflage includes physical, infrared and radar signature reduction)
- mine triggering device

Clear
- or clean appropriate sections of system

Collect
- relevant weather information for the applicable area(s)
- and order and display pertinent information

Communicate
- fire order and other intr-crew instructions

Conduct
- missile system prefire checkouts

Connect
- bridge

Construct
- or assemble bridge

Convert
- transport to launcher

Coordinate
- personnel replacement plans with appropriate organizations

Correct
- applicable defects

Deactivate
- hardware protective device(s)

Decide
- on placement of fire, charge, or pressure in relation to obstacle

Decode
- messages

Destroy

- or alter critical components of communication and other sensitive equipment or documents

Detect

- threat warning(s) which indicate either search or attack modes
- target(s)

Determine

- observable indicators of possible changes in the operational situation
- commander's desired outcome and priorities
- the tactics to be followed
- travel routes for friendly units
- departure and projected arrival times for friendly units
- throughput unit supply requirements
- target type, number, size, direction, speed, elevation
- weather conditions affecting weapons delivery
- target coordinates
- which model(s) of expected enemy behavior best fits collected information
- call signals or frequencies
- which friendly units, with the correct attributes, can be removed from their present operations without unacceptable consequences
- number of targets
- target formation or tactical situation
- the availability of each transportation system required to move each friendly unit and the time required for it to perform its function
- the logistics required by each friendly unit to perform its functions in the operation in question
- the availability of the supplies and delivery systems to the operations area for the required logistics of each friendly unit
- threat potentials of targets
- availability of appropriate friendly weapon system
- the probability of eliminating target(s)
- type of target
- speed and direction of target
- target range at time of weapon delivery
- weather conditions which impact weapon delivery and adjust for them
- type of ammunition to be used based on all above factors
- probable amount of ammunition required to kill target under existing or projected conditions
- effects of fire on target
- the requirements the operation will make on the friendly unit

Develop
- alternative weather projections and their indicators
- policies for area damage control operations
- alternate sources of information

Disassemble
- bridge
- system
- and stow self-recovery components

Disarm
- mine

Display
- all significant information and order it in some logical and helpful manner

Dispose
- of spent casing(s)

Emplace
- system

Encode
- messages

Enter
- communications net

Escape
- from system

Establish
- communications net

Estimate
- casualty rates of friendly forces and projected POW's
- time of arrival and fuel requirements

Excavate
- foundations

Fire
- system
- weapon

Fuel
- vehicle

Guide
- projectile to target

Handoff
- target(s) to attack units
- missile to intermediate guidance

Identify
- friendly unit(s) with the appropriate mix of attributes to match the prioritized requirements
- type and number of potential targets
- threat to system (e.g., onboard fire, flooding, imminent crash, NBC, enemy attack)
- position or route at specified times and locations
- key environmental features
- current weather conditions
- key elements of threat force
- and select routes
- essential information for evaluating NBC contamination hazard outer limits
- appropriate recipients of information
- hazards to movement
- early warning of enemy threat
- critical situations which indicate significant changes in battlefield operations
- present location
- destination
- the nature of the threat(s) from which detected threat warnings emanate
- and determine target coordinates
- target
- important information that is missing
- important information which is internally inconsistent or probably inaccurate

Illuminate
- or designate target

Indicate
- location(s) of forces
- composition (number and type) of forces
- availability of forces
- peculiarities and weaknesses of forces
- recent significant tactical events in which specific units were involved
- actions which forces are currently pursuing (Your consideration of these actions should include direction of movement, speed of movement and apparent purpose(s) of movement)
- the enemy commander's previous behavior in similar situations
- combat effectiveness of forces
- relative threat potentials of enemy forces

- key terrain features which might affect the outcome of the operation (Your consideration of terrain features should include the following: coast-line configuration, exits from beaches, avenues of approach, cover and concealment, observation and fields of fire, defoliated areas, areas suitable for aviation landing, positions for weapons, spaces for maneuver, points of maximum disruption, soil composition, water depth, terrain slopes, beach characteristics, elevations, and accessibility of terrain features)
- man-made obstacles which might affect the outcome of the operation (Your consideration of man-made obstacles should include the following: minefields, tank traps, water obstacles, ditches, and destroyed or potentially destroyed bridges, tunnels, etc.)
- installations which might affect the outcome of the operation (Your consideration of installations should include the following: airports, heliports, enemy depots, enemy command posts, enemy transportation facilities, enemy communication facilities, enemy power operation facilities and lines, enemy C3 positions, enemy air defense systems, enemy radar facilities, and enemy satellite microwave receiving stations.)
- features of weather which might affect the outcome of the operation (visibility data, wind data, temperature data, humidity data, and precipitation data)

Initiate

- firing sequence

Inspect

- system for defects
- mine or triggering device or fusing device
- grenade for defects

Install

- mine (including the digging of a hole)
- sighting components

Launch

- bridge into water
- grenade

Lay

- system for azimuth and elevation

Load

- and secure missile on launcher
- and position cargo and passengers in or on vehicle
- ammunition

Locate
- potential targets

Maintain
- information on maintenance status of equipment needed for mission
- information on current status of supplies

Make
- recommendations about the effects of projected operations

Mark
- target locations; this may be done by physical, chemical, radiological or electronic means

Mate
- warhead to missile

Monitor
- units' compliance with orders and their progress
- intelligence collection and reassign tasks based on updated information
- weather indicator collection and reassign tasks based on updated information

Maneuver
- to protect from threat

Observe
- environment for obstacles, landmarks, etc.

Open
- escape path out of system

Operate
- radar warning receiver

Order
- these requirements based on commander's priorities

Orient
- weapon system in general firing position

Perform
- the following, moving backward (B) or forward(F): Tight turn, wide turn, Accelerating turn, Decelerating turn, rapid acceleration, gradual acceleration, rapid deceleration (no stop), gradual deceleration, sudden stop, maintain constant speed
- takeoff to hover
- instrument takeoff

- hover checks
- hovering turns
- hovering flight
- normal takeoff
- maximum performance takeoff
- straight and level flight
- climbs and descents
- turns
- instrument turns
- acceleration and deceleration
- traffic pattern flight
- high speed flight
- hovering autorotation
- standard autorotation
- standard autorotation with turn
- holding procedures
- unusual attitude recovery
- before-landing check
- shallow approach to a running landing
- landing from hover
- normal landing approach
- shallow landing approach
- steep landing approach
- instrument approach
- GCA approach
- IFR helicopter recovery procedure
- tactical instrument approach
- go-around
- terrain flight takeoff
- hover out of ground effect
- terrain flight navigation
- contour flight
- NOE flight including masking and unmasking
- confined area operations
- slope operations
- pinnacle and ridgeline operations
- evasive maneuvers
- low-level flight
- circling approach from terrain flight
- visual glide slope approach and landing
- ski landing
- amphibious operations
- missile no-go procedure
- misfire procedure
- hangfire procedure

Predict

- maneuver of target(s)
- location of target(s) after given time interval, or predict time interval to arrive at given location (location includes range, altitude, azimuth, elevation, etc.)
- attack of target(s) on frie.ıdly forces
- time and location for successful attack on target(s)

Prepare

- system hardware for obstacle removal or breaching. The nature of this preparation is entirely dependent upon the sort of system under consideration. It may involve preparation for bulldozing, gun firing, demolition, etc.
- contingency plans and the situations in which each is to be implemented
- recovery vehicle
- system to be recovered
- plans, orders, maps and other required documents
- materials for briefing commanders and staffs
- bridge site
- bridge for launching
- personnel estimate based on requirements of operation
- evacuation contingency plans
- system for self-recovery
- mine for installation
- ammunition for firing

Prioritize

- indicators of weather projections
- indicators of operational changes
- recipients for the delivery of information
- pieces of information for delivery
- information according to users' needs and probability of accuracy
- targets
- lists of information users for receipt of information based on their functions in this specific operation and their requirements

Program

- missile

Position
- and emplace launcher
- bridge transporter for launching
- recovery vehicle
- system for escape, if possible under the conditions imposed
- anchors
- system in appropriate location
- sensors in appropriate location

Present
- information about routes which could influence movement

Pull
- system to safe area

Put
- on protective gear and clothing

Read
- and use instruments appropriate to vehicle maneuvering

Receive
- messages

Recognize
- countermeasures and take appropriate action

Recommend
- main and secondary supply routes
- location of rear boundary bases
- movements which are consistent with logistics considerations
- action based on available supply of ammunition, future probable requirements for ammunition, and probable required amount to kill target at various ranges and speeds

Reconnoiter
- recovery area
- for appropriate anchor points and recovery path

Recover
- bridge

Relocate
- target(s)

Remove
- or breach obstacle

Report
- map changes

Secure
- material and cargo for protection against threat
- cargo and passengers

Select
- appropriate location for mine installation
- targets to attack
- target(s) and target order
- type and number of sensors
- designator system position
- the most appropriate friendly unit(s) to engage in operation. (first echelon, reserve, intelligence, counter-intelligence, maintenance, logistics)
- appropriate maps and navigation aids
- travel route
- ammunition

Shift
- to second target

Take
- personal weapon, ammunition, and survival equipment
- appropriate countermeasures to reduce the probability of identification of location (e.g., jamming, smoke, flares, chaff, powered decoys, signature alteration and electronic attach of threat-sensing equipment)

Test
- circuit(s)

Tow
- or lift or push system to be recovered

Transmit
- messages

Transport
- mine

Travel
- designated route

Unload
- vehicle

Update
- plans and orders as battlefield situation changes
- projection probabilities

MANPRINT METHODS MONOGRAPHS:

AIDING THE DEVELOPMENT OF MANNED SYSTEMS PERFORMANCE CRITERIA

COMPREHENSIVE REQUIREMENTS GENERATION CONCEPT PAPER

W.P. Cherry
D. Thompson
L. Proegler
D. Roerty
D. Roberts
Vector Research, Incorporated
P.O. Box 1506
Ann Arbor, Michigan 48106

CONTENTS

CONTENTS (Continued)

LIST OF FIGURES

COMPREHENSIVE REQUIREMENTS GENERATION CONCEPT PAPER

INTRODUCTION

Background

The development of performance requirements and RAM criteria for a new system or for a major product improvement is a lengthy and complex task. The overall responsibility for the task is usually assigned to the appropriate TRADOC school, specifically to the Directorate of Combat Developments. In the context of the Concept-Based Requirements System the development process begins during the Mission Area Analysis and, for all intents and purposes, is completed by the time the Required Operational is issued. For a major system the process typically lasts longer than a year and involves coordination with and the participation of AMC and multiple TRADOC centers and schools as well as virtually all major elements of the Department of the Army.

The objective of this research is to design a system that will support the development of performance requirements and RAM criteria for use by system designers. We have taken this to mean sufficient information must be provided to enable the designer to allocate functions and tasks to soldiers (operators and maintainers) and to hardware/software components and to conduct trade-offs against specified criteria, which enable both the designer and the Army to verify that requirements and criteria have been achieved for the full range of tactical, operational, and environmental conditions in which the system may be deployed. The objective is well chosen. For a variety of reasons it is not unusual to find performance requirements that are, for the system under consideration, internally inconsistent. Similarly it is not unusual to find requirements that are externally inconsistent; that is, criteria applied for a system are inconsistent with the characteristics of the interfaces between that system and other elements of the total force. Requirements and criteria are also often incomplete in the sense that particular sets of tactical, operating, or environmental conditions are omitted or not fully specified.

Complexity is the major contributor to the difficulty of setting unambiguous and objectively measurable criteria of minimally acceptable performance and RAM. Any major system is extraordinarily complex in itself. It must move, operate, communicate, survive, and be sustained in a wide range of conditions: tactical, operational, and environmental. It will contain thousands of parts and numerous subassemblies and subsystems most of which must operate successfully under conditions which are extreme even in peacetime. Added to this is the complexity of the "total system" and activity in which the system in question must operate. Requirements and criteria must be developed to describe all of the activities performed and all of the interactions with elements of the Army, with elements of the threat, and with the environments in which the system must function. In fact, it is virtually impossible to adequately develop the specifications for a particular system without considering how the system will be used in a total force context. It is this fact which in the authors' opinion is central to the development of the product called for in this portion of the research.

The issue of complexity also must be addressed in the context of the combat models that are utilized to analyze and validate performance requirements and RAM criteria. These models range in scope from detailed models of physical processes, such as ballistic penetration of armor or pulse-by-pulse modeling of radar, to models of theater-level joint task force combined arms campaigns. As a general rule complexity is a factor in all; as scope is expanded, level of resolution is reduced. Typically major weapon systems are analyzed using high-resolution models which examine performance over periods of seconds, minutes, or, at most, hours. Because the models must be complex to capture the important interactions which occur they are expensive to employ; consequently analyses focus on a small but significant set of performance parameters under a restricted set of tactical, operational, and environmental conditions. Interfaces with other combat arms, combat support elements, etc., are rarely included. In the larger-scale models these interfaces are included, but in order to achieve the required increase in scope, levels of representation are more aggregated and in many cases high-resolution details of, for example, failure rates, route marches, preventative scheduled maintenance, visibility, range, etc., are analytically subsumed to capture significant features of campaigns involving divisions, corps, and echelons above corps. In spite, however, of the levels of aggregation employed, the level of complexity is high; consequently analyses are relatively expensive in terms of time and resources. Nonetheless, application of the complete hierarchy of models is necessary if requirements and criteria are to be internally and externally consistent and complete. Both high resolution and a total force context must be brought to bear.

The above description of the requirements process is based upon both corporate and personal experience. Most recently, VRI staff have participated in requirements analyses for the Future Armored Combat System, the Armor Family of Vehicles, the Forward Area Air Defense System (both Line of Sight and Non-Line of Sight), the LHX, and the Howitzer Improvement Program. Analyses were (and are being) performed for both the Army and the industrial clients. Based on this experience it appears that the preliminary or initial specification of particular performance requirements is not a problem. Generally speaking the initial values are set to respond to a hypothesized threat and/or to achieve some improvement in system performance. There is a tendency to restrict attention to a system's primary mission and as a consequence its relationships within the battlefield subsystem to which it belongs, and with other battlefield subsystems are neglected. More significantly the consistency of these initial requirements is a problem; i.e., while it is an easy matter to set values, it is difficult to ensure that they meet overall goals in an effective and efficient manner and that they make sense.

There is strong evidence, at least in our experience, that it is not until high-resolution combat models are applied that the implications of different performance requirements are fully understood. This occurs, for example, at the micro level, where detailed analyses of terrain and tactics are necessary to establish engagement parameters and envelopes, including details of timing and geometry. Consideration of broader scope, for example, battalion operations in the context of a division concept, typically leads to a better understanding of tactical movement, target presentation, resupply, etc., and not infrequently to revisions to the Organizational and Operational Concept. The application of high-resolution combat models, with commensurate representations of threat, area of operations, own forces, and tactics and doctrine, thus supports an iterative process in which consistent and logical performance requirements are derived from the initial values.

This iterative process is by no means inexpensive. Use of high-resolution combat models consumes time and resources. However, the level of resolution in models and analysis is necessary. The key is to make the process efficient. Estimates of the number of drafts of requirements produced for a major system range from 15 to 25 at a cost of approximately 500 person-hours per draft. Supporting this process are analyses which can consume from 10 to 20 person-years of effort over a multiyear period. It is this environment in which the product developed in this research is to be employed. A major goal for the product is to make the iterative process more effective and more efficient.

Given that both high resolution and a total force context are required it is useful to consider paradigms which describe the missions or activities of a system. Employing a concept used in modeling one can think of a system and its use as described by a snapshot which contains all the information necessary to describe the state of the system, the length of time it will occupy that state, and the process by which it will transition to its next state. In the context of this research states may be thought of as "missions" for which performance requirements and criteria are required. Clearly occupancy times and transitions to "next states" are influenced by interactions with friendly forces, threat forces, and tactical, operational, and environmental conditions. Abstractly the development of a complete set of requirements and criteria depends upon the creation of a full, high-resolution state space and an understanding of the necessary or likely occupancy times and transitions. Conceptually the developer would like to have a complete set of trajectories through the state space, a complete set of possible system "histories." The process of defining performance requirements would then center on naming each state or mission, specifying the conditions under which it occurred, examining all appropriate interfaces to all other systems (the total force), and specifying required performance requirements and criteria. Most of the higher-resolution, large-scale combat models are based upon this concept. The key to their use in insuring completeness and consistency depends upon the extent to which they represent the entire combined arms process, and the resolution with which they represent the missions of a system and the conditions under which they are performed.

Basis for Development

As noted in the preceding section the principal problems encountered in developing system performance requirements are those of consistency and completeness. Consistency can be both internal (e.g., performance requirements for system subfunctions are in conflict or not balanced), or external (e.g., performance requirements for the system fail to take into account the nature of its interfaces with other systems and with other battlefield functional areas). Completeness of a set of system performance requirements implies that all missions and interfaces have been specified. The key issue in developing requirements thus is one of adopting a structure that ensures that all missions and interfaces are considered. Note in this regard that the "missions" of the theater, army, corps, division, etc., are not determined in the process of identifying performance requirements, rather they must be considered to ensure that all requirements are set. In particular, activities in which a system participates because of corps or division missions may have little to do with its primary functions but have a major impact on its contribution to overall force effectiveness.

A number of years ago GEN William E. DePuy (US Army, Retired) introduced a structure, similar to that illustrated in figure 1, to serve as a basis for describing the processes of synchronization and command and control. The DePuy structure is a matrix in which the rows correspond to echelons and the columns to major battlefield functional subsystems. In the context in which the structure was originally used the emphasis was on command and control. Horizontal lines represented coordination of battlefield functional subsystems; vertical lines represented "stovepipe" command and control. The DePuy matrix serves as a basis for a different interpretation, namely, one in which the emphasis is on requirements.

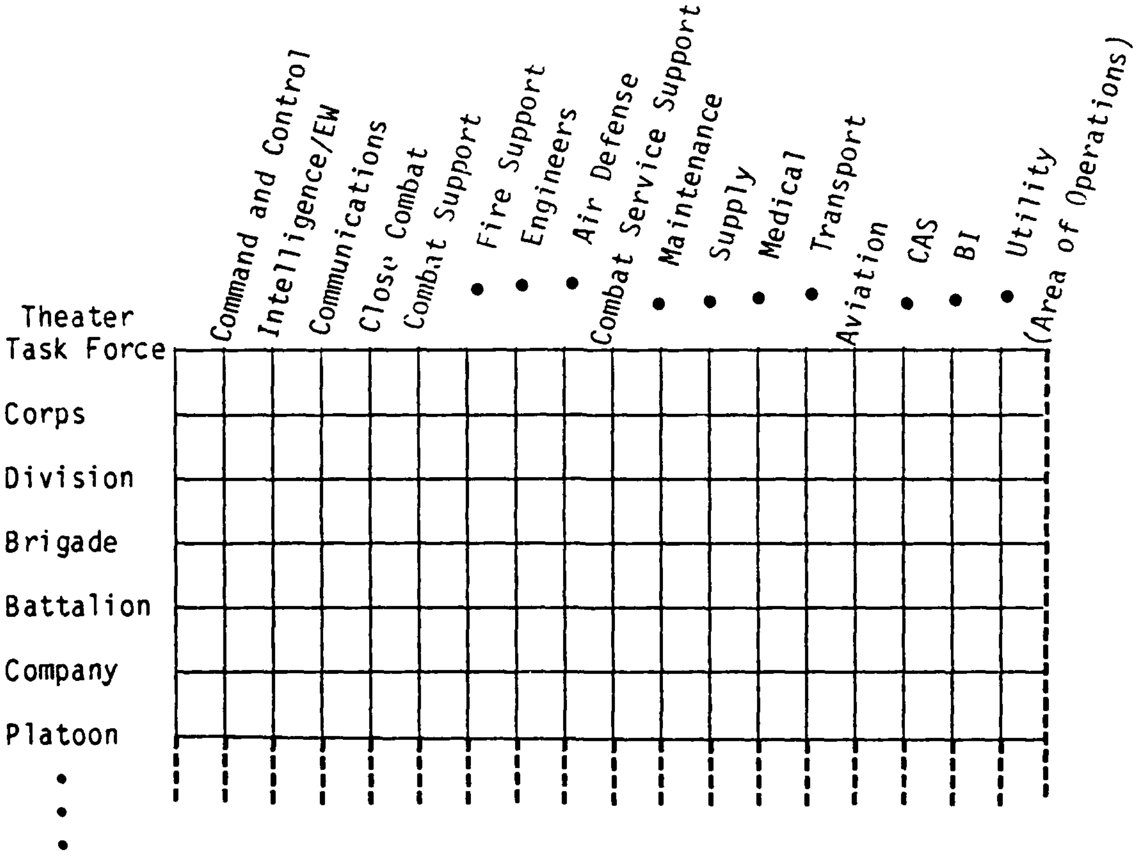


Figure 1. Echelon/functional organization for requirements generation.

Any system interacts to some degree with every node in the matrix. For example, a tank fought at the platoon or lower level must be transported at the joint task force or theater level. If it is to be air transportable it must conform to certain size and weight constraints. At the corps level the tank can move under its own power, but may utilize railway transportation or heavy equipment transporters, both of which lead to size and weight constraints. At the division level road marches are more critical, while at company and below cross-country mobility is important. Finally, at the level of the squad and single tank agility becomes the prime issue. Clearly each of these "linkages" is subject to conditioning imposed by the area of operations, conditioning that becomes progressively more detailed (climate, soil, cover, slope, etc.) as one progresses down the rows of the matrix.

While the above example tends to focus on interactions with the total friendly force, the matrix also provides a basis for organizing interactions with the threat. Consider, for example, a command and control subsystem located at the division echelon. Its interaction with the threat close combat subsystem should be minimal, but may include establishing perimeter guards to detect and (possibly) deal with threat rear area operations. Its interaction with the threat intelligence and electronic warfare system will lead to requirements associated with its vulnerability to detection, classification, and location and contributes to requirements for a jump capability. Its interaction with the threat fire support system similarly leads to requirements for ballistic, nuclear, biological, and chemical protection.

Based on our assessment of the overriding need for completeness, we have adopted the matrix approach to organize the process by which performance requirements will be generated. In essence, two matrices will be used: one for the "own force" interactions and one for the "threat force" interactions. Environmental and other conditions associated with the area(s) of operations will be included via a specific column in the "own force" matrix.

To further assist in the process of identifying performance requirements and criteria we will employ a categorization of time and space into an Theater or Army year, a Corps campaign month, a Division week, a Battalion day, a Company/Battery hour, and a System minute. We believe that this set of categories will facilitate a top-down approach to developing a complete, high-resolution set of missions; i.e., activities involving the system or functions which it is performing, the conditions under which performance occurs, and the interfaces with other elements of own force and the threat force. These activities or functions may range, for example, from intratheater transportation via rail, through annual scheduled maintenance at intermediate direct support, march to the line of departure, to firing six rounds of DPICM at an extreme range target in one minute at night, in a clear visibility situation, on a reverse slope of 10°, in 3 °C weather.

The development of procedures to ensure completeness centers on identifying missions and the interactions that take place between nodes. The problem of ensuring consistency will center on specifying mission standards taking into account the interactions. As in the case of completeness both own force and threat force interactions must be addressed.

For example, the rate of fire of an artillery piece is a major design driver. It is related to the number and rate at which targets are supplied by the close combat subsystem and the intelligence/EW system, and in turn must consider on-board ammunition and ammunition resupply capacity. Over a longer term the rate of fire contributes to operational equipment failures and RAM. Similar interactions must be addressed for a close combat system in a direct-fire engagement. Lethality involves trade-offs between range, acquisition, accuracy, firing rate, probability of a kill given a hit, etc., design attributes which can only be set after considering the threat and the area of operations. These parameters also interact with issues of agility and mobility. For example, in the case of prepared defensive positions for hull-down fires, what are gun elevation-depression requirements and how do these interact with the area of operations and the capacity and capability of engineer systems?

Establishing RAM criteria is not a single system issue but also requires use of a total force perspective. From a capacity perspective maintenance is provided (in varying degrees) by the crew, the unit, intermediate support, and depot. Maintainability is itself a function of system design. From a reliability perspective, failure rates depend on how and where a system is used or, equivalently, mission activities and frequencies under tactical, operational, and environmental conditions. Mission activities and frequencies are determined by the capacity of the basic organizational unit, for example, a battery or section. This capacity to service demand is at any instant a function of the number of systems assigned, their individual capabilities, and their operational availability. From a single system perspective it is possible to establish a pseudomission defined to be repair of failure or scheduled maintenance and to specify its acceptable duration under a set of conditions. (Interfaces to the maintenance subsystem would necessarily be defined and considered.) This would be directly related to operational availability of the system, and, by considering the complete range of missions, reliability criteria could be established. However, that criteria is only valid if during the repair/maintenance submission the total force has the capacity to service the particular demand in question. In the case of a 155mm self-propelled battery in direct support this capacity could be provided by the battery, by another battery of the battalion, or by a battalion in general support. For RAM criteria alternative approaches must be provided depending upon whether or not the number of systems in the basic organizational unit is specified in advance. Given this number, the demand process (or, equivalently, mission frequency), and the total service capacity, operational availability and thus maintainability and reliability criteria can be derived using system performance requirements and capacities. In the absence of the number of systems in the basic organizational unit, more comprehensive analyses of numbers, capabilities, and operational availability will be required.

Quantitative descriptions of performance requirements generally are addressed using combat models. These range in detail from the detailed high-resolution models, such as those that address ballistic penetration of armor or individual radar returns, to more comprehensive models addressing duels, battalion task force engagements, division and brigade battles, and corps- and theater-level campaigns. These models are typically expensive and time consuming to employ and do not uniformly use measurable inputs, nor do they uniformly represent all of the vertical subsystems and echelons represented in the DePuy matrix. Nonetheless, they constitute perhaps the only methodology in widespread use to evaluate and trade off different levels of performance requirements. The key to obtaining consistent (and complete) requirements is knowing what questions to ask of what models. Because of the level of detail inherent in the high-resolution, narrow-scope models and because of the complexity inherent in the lower-resolution, broader-scope models, both of which are necessary, the development of performance requirements involves a large number of experts and consumes reasonable amounts of time. Thus, the development of complete and consistent requirements should be viewed as a process of management and coordination and not as a process performed in a short period of time by a small group of personnel.

Structurally, the development of complete and consistent requirements can be resolved into:

1. Providing a total force overview that ensures that a system's activities or missions and interactions with all elements of the force are identified.

2. Providing a similar overview of the threat force and area(s) of operation that ensures that a system's interactions with all the elements of the threat force are identified.

3. Providing a means by which the quantitative statements of performance, which are derived from the interactions, are consistent and rational.

An analogue of the DePuy matrix will be used to organize the identification of missions and interactions. To facilitate the development of missions statements, to identify requirements, and to provide the rationale for a categorization of time into Theater/Army year, Corps campaign month, Division week, Battalion day, Company/Battery hour, and System minute will be utilized. In concept the two structures are designed to produce descriptions of a system's "life" or a complete set of missions and mission sequences defined in high resolution. Quantitative statements of performance requirements and criteria will be derived primarily by use of combat models and analysis with consistency provided by identification of the quantitative aspects of the interactions between the system, the remainder of the friendly force, the threat force, and the environment for each mission undertaken. The process of setting quantitative requirements will be iterative, proceeding from initial values set by the combat developer.

Outline

The remainder of this paper consists of two sections. The next section provides an overview of the major components of our concept and is organized as follows: provides an overview of the concept, presents an example of use of the system, and addresses output, knowledge base, and data sources, respectively. The last section describes the development and deployment of the aid.

DESCRIPTION OF THE CONCEPT

The purpose of this section is to provide a preliminary description of the salient details of our concept for a product which will support comprehensive requirements generation. The product will be an expert system, which is designed to ensure that performance requirements and RAM criteria are consistent and complete and unambiguously, objectively measurable. It is intended to provide expertise and support to the combat developer to ensure that all requirements and criteria are specified. By itself it will not set the initial quantitative values of those requirements and criteria. It will provide "expertise" in setting such initial values and then support an iterative process of refinement.

Overview of the Concept

Our concept for the aid is an expert system, with a subject area of expertise in Army system requirements, to guide the user through the process of identifying, defining, and setting levels for a complete and consistent set of system requirements. This approach is derived from our understanding of the difficulties with the current requirements process, the constraints of time and resources placed on the requirements analysts, and of the kinds of information that really should be considered in requirements generation. This section describes the critical issues involved in the design of the aid, our response to these issues, and the overall nature of the aid entailed by our response.

This section states the purpose of the aid; i.e., the kind of assistance that it is to give to the requirements analyst. It summarizes the difficulties with the current methods of setting requirements and highlights the features of the requirements process where the aid can alleviate the problems, and describes our concept.

Purpose

The purpose of the aid is to supply the requirements analyst with advice and expertise, otherwise not available to him, that will guide him to the production of a complete and consistent set of requirements. To do this, the aid must assist the analyst in gaining an understanding of the operational implications of system requirements. This means that the aid will guide the user through the process of identifying and setting requirements, understanding the reasons for the requirements, and identifying links to combat models and other sources of information for setting criteria levels. The principal reason for this approach is the need on the part of requirements analysts to understand the full range of missions and functions that systems must perform, the full range of operational environments in which their missions and functions must be performed, and the relationships among the various requirements. Without this understanding there is no guarantee of completeness and consistency of requirements.

Where Improvement is Needed

Our approach to the aid derives from the fact that the many difficulties encountered in the requirements process stem from three general problems: an incomplete perspective on the operation of the system, an incomplete perspective on the place of the system in the force, and the setting of inappropriate levels of performance on those criteria that are considered. These problems arise from identifiable characteristics of the way that the requirements process is carried out, mostly relating to limitations on tne quantity and quality of information available to requirements analysts. By helping the analysts to consider aspects of the system and its role within the force that would otherwise be ignored, the aid can lead him to a more complete and consistent set of requirements, and by informing him of links to combat models and other analysis tools it can help him to develop criteria that are realistic and supportable.

Problems with the completeness of requirements stem from failures to take a total system perspective and a total force perspective. In the present requirements process the greatest emphasis tends to be on the functions that are thought of as constituting its primary role, such as the target-servicing function of a weapon system with the associated performance criteria of accuracy, lethality, and rate of fire. Attention tends not to be given to the functions of the system that, from this point of view, are seen as secondary to the primary mission, even if they are necessary to its performance. This limited perspective stems from several circumstances in the Army's requirements process. Time pressures of analyst's work environment, the kinds and levels of experience of analysts, and the nature of the tools available to the analysts all tend to constrain any attempt to consider a complete set of requirements, and the general focus of the requirements process, which is on the materiel solution to a deficiency, does not encourage attention to aspects of a system unrelated to the materiel solution. Our aid, by guiding the analyst to an overall understanding of the system and its role within the force, will provide the means to overcome these constraints.

First, the schedules for requirements generation leave little time for consideration of all possible relevant requirements. An automated aid can, at the least, encompass enough expertise in Army operations that it can present the options quickly and completely and assure that analysts have the opportunity to apportion the available study time to them. In addition, by directing the user to available sources of data (e.g., by identifying the link between a requirement and an Army data base or a specific output of a combat model), it can speed the acquisition and use for the data.

Second, for those personnel who lack the operational Army experience to understand the system and its overall role within the force, the aid will fill the gap in their expertise by drawing their attention to the complete set of relevant criteria on which requirements need to be set. By explaining the reasons for setting these performance criteria it can assist them in understanding the importance of each requirement, and it can thereby enhance their overall judgment of the problem and their ability to judge the relevance of each requirement to the particulars of the system under consideration. For those analysts who lack the experience of another sort -- with combat models and other formal analysis tools -- the aid will supply advice in that area of expertise by identif.'ing the tools that can be used to derive values for a performance criterion. In addition, the Army maintains numerous data bases in different locations about many different aspects of systems and forces, and even experienced analysis personnel may not know where to obtain a particular item -- another area where the aid could assist.

A third way in which the aid can help concerns shortcomings with the models and other tools that can be used in the process of setting requirements. No tool such as a combat model or logistics model, for example, can contain a complete representation of military operations, and this incompleteness tends to focus attention on those aspects of a system that are represented explicitly in the tool. In fact, much study effort often goes into alleviating such shortcomings of incompleteness with models through such steps as modifying a model to make it more complete or doing external side analyses to manipulate its inputs or outputs. If an analyst's expertise does not extend to a particular kind of tool, it is important for the aid to inform him of the particular links that can be made to the model, or to warn him when the links are missing and inform him of an external source of data to be consulted or of a side analysis that should be performed.

By addressing all of the above-listed problems in the requirements process our aid will assure that consideration is given to the full range of performance and conditions necessary to guarantee that a developed system will operate as expected. In addition, by considering the relationships among requirements it will assure that requirements are consistent, and by guiding users to the links with combat models and other sources of information it will help increase the chances that the required levels of performance are objective and defensible. The next section presents an overview of how the aid will accomplish these goals.

Description of the Approach

The overall approach of the aid is a top-down analysis and hierarchical decomposition of the performance characteristics of the system under consideration. This approach is aimed to remedy the primary shortcoming of the requirements process, which is the lack of a total system and a total force perspective, and it does this by incorporating expertise about generic classes of Army systems, which it uses to guide the user through a structured exploration of the possible requirements. To do this the aid must perform three basic functions: (1) act as a source of knowledge about the system under consideration, (2) use this knowledge in a structured way to explore the set of requirements for the system, and (3) support the user as it leads him through this exploration. These functions are basic to the functional description of the aid, and they underlie the approach described here.

To develop a complete and consistent set of requirements for the system, the aid utilizes a top-down analysis and hierarchical decomposition to define the system missions for which performance and RAM criteria are to be defined. Defining the missions means examining the interfaces among the functions that the system must perform, the functions of the total force, and the conditions under which the system and the force operate (i.e., the threat, the physical environment, the tactics). By exploring this set of interfaces, the aid can ensure not only that the total set of missions is complete and consistent, but also that each is defined in enough detail that developers can objectively determine if a system will satisfy it. That is, the aid decomposes a mission into enough submissions and missions and factors, defined in sufficient detail, then the requirement is defined unambiguously.

The information necessary to explore all of these interfaces is embodied in the aid's knowledge base and is selected by the aid when the user specifies the generic class of system under consideration. As each mission becomes fully defined (after all the interfaces pertaining to the mission have been identified) and confirmed as relevant to the system under consideration, the aid offers additional advice to assist the user in selecting a level of performance as the requirement for the mission. Several kinds of advice are needed at this point, including sources of information (e.g., links to combat models), interpretations of the performance measure for the mission and explanations of the reasons for placing requirements on the mission, and the identification of related requirements that should be checked for consistency.

The process of top-down analysis and hierarchical decomposition of the problem is guided by that portion of the knowledge base pertaining to the class of system selected by the user. The top-down analysis relates to the differing requirements on a system as viewed from the perspective of different force echelons, from theater down through corps, and so on, to the squad, section, and the system itself. At each echelon there arise different considerations relating to the functional areas of the system that are of importance, the relevant characteristics of the theater of operations, and the interfaces with friendly and enemy forces that impact on the selection of performance requirements for the system. The analysis tools that can be used for setting performance levels for requirements also vary from level to level, and different kinds of links to these tools will occur at each level.

The theater-level echelon at the top provides a perspective emphasizing characteristics of the system that influence its intertheater deployment, for example. Below that, the corps-level echelon provides a perspective on such issues as the corps-level components of the maintenance system, intratheater movement, prepositioned stocks, and corps-level fire support assets and intelligence assets. At the division level attention is focused on those echelons of the maintenance system, on tactical movements, direct support artillery, and associated elements for command, control, and communications. At the battalion level are such processes as unit-level maintenance, higher-level resolution of tactical movement, and interfaces to other elements of the combined arms team. At the company, squad, and section levels these issues are encountered at a still finer levels of resolution and with emphasis on the functions related to the primary combat-related roles of small units. At the lowest level of the hierarchy is the individual system for which the issues of importance are such missions as the execution of fire missions, one-sided and two-sided duels with targets, and the processing of target acquisition information.

At each level in the hierarchy different functions of the system are performed, different conditions influence the performance, and there are interactions with different portions of its own forces and enemy forces. The span of time over which missions are defined becomes increasingly shorter at the lower levels of the hierarchies, and the set of analysis tools is different at the various levels. This means that the set of linkages to the analysis tools varies with the echelon as well as with the particular function and system, and these differences will be recognized by the aid. One important class of tool is the combat model, of which important examples for the Army are FORCEM at the theater level, CORDIVEM and VIC at the corps and division levels, FOURCE at the division level, and CASTFOREM and CARMONETTE at the battalion and company levels. At the system level models tend to be much more special purpose in their nature, such as the variety of engineering performance models and operator workstation models. With even this partial example of tools that can be needed in the requirements process, it is clear that a great deal of time can be saved by an aid that automatically leads the user to the analysis tools that can address his requirements.

In the introduction to this paper, echelon was one of the dimensions of the matrix presented as an organizing framework for requirements generation. As utilized in the aid, the echelon dimension of the organizing framework leads, as discussed above, to an enumeration of the functions that the system must perform, the conditions under which it performs them, and its complete set of interfaces with other elements of the total force, friendly and opposing. Once the top-down analysis has defined a mission to which requirements criteria should be attached, the requirements for that mission can be completed by the process of hierarchical decomposition. The purpose of hierarchical decomposition is to ascertain the functions that the system is to perform (move, shoot, sense, etc.) in the mission, and to complete the description of the conditions under which the functions are performed.

It is at this point that the mission becomes defined completely enough to allow for setting quantitative requirements, and the aid completes the process of hierarchical decomposition by supporting the establishment of the value of the required permanence level. Several kinds of information are in the knowledge base to aid in setting requirements levels. For initial values the aid will contain predecessor data derived from current Army systems. These will be described in objectively measurable terms. The aid will also provide guidance on the implications of changing those values. As mentioned above, the aid identifies the links to analysis tools that can be used to iteratively analyze and set the requirement. The name of a model, the related inputs and outputs, and the identity of a responsible organization are examples of this link. The name and location of an Army data base is another example. In addi tion, it is necessary for the requirements analyst to appreciate the reasons for the requirement if he is to set reasonable values, and the aid supports his understanding by explaining the reason for the inclusion of the mission on which tne requirement is to be placed and by identifying the conditions that influence the accomplishment of the mission. The aid also identifies related mission so that the total set of requirements can be set consistently.

With the completion of hierarchical decomposition, the matrix presented in the introduction is completely implemented. The top-down analysis has explored the dimension of the echelon levels from the theater to the individual system and has identified the missions that the system must perform. The aid's process of hierarchical decomposition has explored the functions which the system must perform to accomplish each mission, and in doing this has explored the dimensions of the matrix that correspond to the various battlefield systems, friendly and opposing, that exist at that echelon and to the interfaces with systems to other echelons of command. In this way the information in the knowledge base of the aid has guided the user to a complete and comprehensive set of system requirements and has advised him on setting the values for those requirements.

Illustration of the Concept

The important part of our concept for the aid is not in the software aspects of the design (expert systems and shells for creating them exist already in a variety of forms), but rather in the content of the knowledge base, most specifically in the tailoring of the knowledge base to provide expertise in the unique subject matter of requirements analysis. For this reason it is important to understand the kind of support that the aid will render to the requirements analyst and how the aid will appear to the user during the process of working through a requirements problem. To help in appreciating precisely what is being proposed in this paper, this section describes examples of how sessions with the aid would proceed.

There are two examples. One illustrates the top-level interactions that would occur when one has a new requirements problem, sets it up on the aid, and identifies the broad mission areas where requirements are needed. The other example picks up the user at a subsequent stage of the requirements analysis, after the system and its missions have been broadly defined, and the user is faced with the problem of defining criteria for the performance of some specific functions.

Top-Level Setup

The example below illustrates the exchange between the aid and the user when the process of establishing requirements is first initiated for a new system. The user will identify the new problem and use the aid to identify high-level definitions of missions for the system that he will explore in more detail later. The example dialogue is presented in two columns, with dialogue described on the left, and commentary on the right.

Where sample dialogue is given, the aid's prompts and responses appear in lower case, and the user's entries in upper case. It is not meant to suggest that the precise appearance of aid prompts and user responses will appear as shown below. The aim of the example is to illustrate the content of the session and not the form of input and output on a display screen or other devices. Choices of methods of interactions (menus, commands, etc.) and their precise forms are matters for more detailed design.

DESCRIPTION OF DIALOGUE	COMMENTS
new system or old? NEW	The aid first establishes whether a new problem is to be initiated or an old problem is to be retrieved for completion or amendment.
select a class to which the new system belongs: air defense armor ARTILLERY communications engineer aviation	The aid presents the user with a list of system classes for which it possesses knowledge bases, and the user selects one. In this case the user has chosen to define a new artillery system.
at which echelons is the system used?	The user is prompted to select the echelons at which the system is used and the role which it will fulfill at each echelon -- direct support, general support, or both. After this question is answered, the production rules generate the appropriate set of missions necessary to provide a complete description of a system of this class.
identify the missions[1] that apply to the system:	These missions apply in general to systems of this class, and not all of these missions may be needed to specify requirements for the current system. Some missions may involve essentially the same functions under such similar conditions that they need not be considered separately.

echelon	ds	gs
theater		
corps		
division	X	X
brigade	X	X
etc.		

mission	chosen
attack targets of opportunity	X
attack requested targets	

[1] These missions are prime or operate missions. The knowledge base would also present missions under move, communicate, survive, and sustain categories. The missions shown represent the set of potential missions derived from a single collective mission (e.g., provide direct-fire support) associated with a single mission category (e.g., performance).

DESCRIPTION OF DIALOGUE

provide final protective fires	X
provide counterbattery fires	X
provide illumination	
emplace mines	
deliver chemical munitions	X

next you need to define the functions that the system must perform in order to accomplish the chosen missions.

select one of the missions and you will be provided with a suggested list of functions for the mission profile:

COMMENTS

In this case the user has selected only some of the missions for which the knowledge base contains rules for specifying requirements. For a system of a different class the aid would pose a list of a different set of missions, for which it would possess a different set of rules.

The aid now starts the user on the process of defining the missions that he has selected. It starts by presenting him with templates of the functions that must be accomplished in order to perform the mission successfully, and it will later help him in defining all of the conditions that influence performance.

mission: ATTACK TARGETS OF OPPORTUNITY

mission: attack targets of opportunity

performed at echelons:

division
brigade

please identify functions for this mission at echelon = division

The aid's knowledge base supplies from its base of generic requirements data a template that lists the functions that must be performed to accomplish the mission at the specified echelon.

The user is given the option of modifying the function template based on his understanding of how this particular system performs the function.

function	status
relocate system	... PRECONDITION
navigation	... FUNCTION
acquire target information	... FUNCTION
establish communication with FDC	... DOES NOT APPLY
fire the mission	... FUNCTION
respond to category F2 failure	... FUNCTION

Here the user indicates that one function "does not apply," perhaps because the system operates autonomously and does not require the performance of the communications task with a fire direction center (FDC).

Four of the functions are marked as such by the user, by which he indicates that he wants the aid to develop the requirements criteria for them when it gets to that stage of the problem.

DESCRIPTION OF DIALOGUE

COMMENTS

Other functions may contribute to mission accomplishment, but the user may prefer to direct the aid to deal with some functions elsewhere in the problem and not to repeat that stage of the problem within this mission.

In that case he marks a function as a "precondition" so that the aid will know that it is still part of the template for the mission profile and is to be maintained as part of the context for the performance of the other functions of the task. If the user instead marked the relocation task as "not applicable," he might be reminded later on that the level of performance of that task was a candidate for inclusion in the set of overall conditions for performance of other mission tasks, and he changed it to a precondition.

The user might avail himself of various help features during this process, for example, asking for definitions of the functions or requesting to view the functional templates for other missions. He should also have assistance in understanding the choices that he can make in altering the template, and he should have the option of seeing the choices explained after he has made his selections.

At any stage in the problem the user would be able to view summaries of the problem's current state of completion. The following table would tell him that he has chosen to work with four of the seven possible missions and has not yet defined the requirement for performing any functions. Note that a reduced number of functions appears for the mission that the user has just modified.

DESCRIPTION OF DIALOGUE

here is a table of the missions that this kind of system must perform, showing the numbers of functions involved in each and the number of functions for which you have completed setting the requirements:

COMMENTS

If the number of missions were unmanageably large, the aid would group them into collective missions and present a similar table for each collection, so that the user could work down to the level of the mission.

	number of functions in mission			
	division		brigade	
mission	total	complete	total	complete
attack targets of opportunity	4	0	4	0
attack targets requested				
provide final protective fires	6	0	6	0
provide counterbattery fires	6	0	6	0
provide illumination				
emplace mines				
deliver chemical munitions	7	0	7	0

Setting Performance Criteria

After mission profiles have been defined, as shown above, the user should have the option of proceeding in any of several ways. He should be able to review and modify what he has done, to receive an explanation of the implications of his choices, or to proceed to specify performance criteria. If he proceeds to specify the criteria, he should also be free to explore the set of criteria in a depth-first or breadth-first manner, depending on the procedures that he wishes to follow. The performance requirements for all functions of a mission might be completed before proceeding to another mission, or requirements might instead be set for all occurrences of a function in all missions before proceeding to another function. In either style of work the aid will eventually enter into a dialogue (such as the following example) to help the user set the performance criteria for performing a specific function within a specific mission. The example begins after all of the necessary context about the mission and echelon has been established, and the task is to define a specific occurrence of the function; i.e., all of the conditions that influence the selection of the criterion level and the system's ability to achieve the level.

DESCRIPTION OF DIALOGUE

COMMENTS

function = establish communications
with FDC

it is part of the mission:
deliver chemical munitions

DESCRIPTION OF DIALOGUE	COMMENTS
do you wish to look at the other functions of the mission? NO	
do you wish to look at requirements already established for other occurrences of this function? NO	The user may already have developed requirements for this function for this mission or for another mission. In either case it might be useful for him to examine the prior case. If done for another mission, he may have specified the same conditions for its accomplishment that he will specify in this case, and the two requirements may be redundant. If done for the current mission under different conditions, the user may want to ensure consistent trends as the performance conditions are varied.
the performance measure for this function is: the probability of successful message transmission within a specified time interval	Before specifying the detailed conditions for performing the function, the aid defines the quantitative measure on which a criterion level is to be placed.
the ability of systems of this class to perform the function depends on the following conditions: tactical conditions range to the FDC, etc. operational conditions background message intensity, etc.	The full set of conditions on which function performance is dependent for systems of this class in the current mission context is then enumerated by the aid to prepare the user for the process of identifying the conditions in which the current requirement is to be placed.
threat elements ECM equipment, etc. other system functions availability of communications subsystem, etc.	The aid continues the list of influencing conditions, of which examples are shown here. It will then provide the user with typical ranges of values and prompt for the the values to be used in specifying the current requirement.

DESCRIPTION OF DIALOGUE	COMMENTS
please specify the background message intensity (messages per minute) and the average message duration (minutes): typical range in mid-intensity warfare is an intensity of 1.2 to 1.8 and an average message duration of 0.4 to 0.5	In working with the user to specify a complete and unambiguous set of conditions for performance of the function, the aid should be able to provide assistance in the form of guidance on the impact of each influencing condition on the performance of the function.
WHY?	Here the user asks for information on the impact of traffic intensity and message duration.
the performance measure for the function is: the probability of successful message transmission to the FDC in 30 seconds	After the conditions have been explained to the user and their values have been defined, the guide assists the user in setting the value of the required performance.
sources of information on the probability of successful message transmission can be obtained from (a) queueing models of the communications network and (b) models of radio wave propagation needed to provide some of the inputs to the queueing models.	Its guidance includes identifying links to sources of information from which values for the requirement can be set.
WHERE? for inputs to the queueing models, the IEW Functional Area Model at TRAC-WSMR will supply the probability of propagation	Here the user asks the aid to be more specific as to the information source, and it responds by identifying a specific model and the link between the tools that the user must deal with.

Output

The aid will generate many kinds of outputs to the user and to physical devices. From the user's point of view the important outputs of the program consist of the information presented to him as assistance as he works through a requirements problem and the collection of defined requirements that the aid collects along the way. The ultimate output of the process of using the aid is the latter kind of information, the system requirements, along with such supporting information as the reasoning behind the requirements and links to analysis tools from which values are to be set. This section discusses the logical content of the major outputs needed to define and justify requirements and leaves for the stage of more detailed design the definition of screens presented to the user, output to printers and electronic storage devices, and other details. It also leaves for more detailed design the discussion of outputs associated with the process of using the tool to update the knowledge base to define new categories of systems or update production rules.

The outputs that are central to defining system requirements are listed below:

- indexes of categories of problems for which rules exist;
- the production rules and supporting data; and
- partial and finished problems:
    - index to the set of problems,
    - set of alternate versions of a problem,
    - collection of requirements of a given category,
    - an individual requirement for a system, and
    - additional elements needed to define a requirement.

The list above distinguishes three general kinds of outputs: outputs that act as indices and guide the user to the information available on the system, outputs that describe the production rules and supporting data that the aid uses to solve a requirements problem for any system of a given class, and outputs that describe the set of requirements that have actually been defined for any problem that has been solved fully or is in the process of being solved.

The aid will combine these basic kinds of information in different ways, depending on the job that it is doing. Examples of the various kinds of interactions that the aid may have with the user, and for which it will combine this information in various ways as outputs are: informing the user of the kinds of systems for which it possesses production rules, explaining the rules that it uses for a given class of system, explaining the reason for the inclusion of a rule, leading the user through a problem, assisting him in defining a specific requirement for that system, or reviewing the completed requirements, or making a comparison of requirements to assure consistency in their levels of required performance.

The aid will be capable of handling problems (i.e., the problem of defining the requirements for a given system) for different systems and must be able to support different versions of the analysis of a single system. For example, archived copies of a problem in different stages of completion would constitute different versions, as would parallel studies performed under different assumptions of the future environment of the conflict, the threat, or the degree of technology opportunity. The aid must provide output to support the user in enumerating and locating the problems that are accessible to the aid.

For a problem in any state of completion the aid must maintain an associated data base of the completed requirements. For purposes of output the description of a requirement consists of the information necessary to define its performance criterion, along with contextual information provided by the aid to assist the user in uderstanding the requirement. The latter kind of information is supplied by the aid and consists of such items as links to combat models and logic traces linking the requirement to the generic knowledge in the knowledge base. The purpose of these logic traces is to tie the requirement to the production rules and generic requirements data of the knowledge base, such as the definitions missions and conditions for their performance. Such ties are a necessary part of the aid, so that on output it can present a complete definition and justification of a requirement to the user. The presence of these ties to other parts of the data base also reflects the linkages that the aid must maintain in the knowledge base while working through a problem. Provisions will also be incorporated so that the user can review the iterative process by which values have been set; i.e., the aid will describe what should be done and why, and maintain a trace of the salient details of what has been done as specifications become more complete and consistent.

Output that the aid must provide concerning a single requirement consists of the following kinds of items:

- definition of the performance measure;
- units of measurement;
- numerical value requirement, if one has been established;
- qualifiers for achievement of this level; e.g., frequency of occasions on which it is to be met, or confidence interval;
- description of the mission;
- trace of production rules and generic requirements data;
- explanations of the requirement (reasons for its inclusion); and
- links to analysis tools.

The first four items of the list are necessary to specify the requirement rigorously. They define the kind of performance required of the system, how it is measured, the level of performance required, and any additional information needed for multidimensional measures. The fifth item, the description of the mission, provides the context for the performance and contains the following kinds of information:

- general nature of the scenario:
    - opposing forces,

    - friendly forces,
    - theater of operations; and
- specific mission profile, if needed;
- system functions involved;
- friendly force elements interacted with;
- threat force elements interacted with;
- relevant characteristics of the physical environment; and
- tactics used.

A trace of production rules and generic requirements data involved will be useful in explaining to the user why the requirement is being considered and how it is related to other requirements with which there may be a need for consistency.

Associated with every requirement, the aid should produce as output certain guidance on its interpretation. To some extent a trace of the production rules and generic requirements can help in this interpretation, but in some cases it may be necessary to include in the knowledge base some text, associated with the requirement, that explains it. For example, a size requirement may follow logically from a set of production rules that apply to the intratheater transport of the system, but the requirement will make more sense to the user if the aid explains in addition that the size requirement applies if the system is being airlifted in-theater and is one of the constraints that must be met if the system is to be carried in a C-130 aircraft. The output of the aid should also associate each requirement with sources of data, such as links to combat models.

Knowledge Base

The purpose of this section is to discuss the information that will be contained in the knowledge base. The purpose of the knowledge base is to provide the expert knowledge required to develop system performance requirements as well as to record user decisions as requirements are developed for particular systems. In some sense, this information may be more or less homogeneous depending on the approach selected: in an OPs-5 environment the discrimination would be between the working set and production rules; in a Prolog approach, rules and data would be essentially indistinguishable. We are not presently committed to a particular expert system approach, but, for purposes of exposition, have chosen to discuss three principal classes of knowledge base content. For the purposes of this discussion, the knowledge base content can be partitioned into three classes:

- generic requirements data, for example:
  - information concerning the types of missions to be considered when defining the required missions for individual major systems; and
  - information concerning the types of conditions and range of performance requirements necessary to specify each type of mission;
- specific system data, for example:
  - information describing the system in terms of type, functional description, potential employment;
  - information concerning the required missions currently selected for a specific system;
  - information concerning the types of conditions and range of performance dimensions selected to define mission performance for each of the missions selected; and
  - information concerning values specified for each condition set and performance dimension selected; and
- production rules to assist the user in developing specific requirements, for example:
  - identifying and selecting missions appropriate for a particular system;
  - identifying and selecting types of conditions and range of performance dimensions for a particular system and mission;
  - selecting specific values for condition sets and performance dimensions selected;
  - insuring consistency among performance criteria specified; and
  - insuring completeness of system specifications.

The generic requirements data provide the framework, or context, for the development of specific system requirements while the production rules provide the expert knowledge to assist the user in the development of a complete and consistent set of performance specifications within the context provided. Data concerning specific requirements is created or modified as part of the development process, recording specific user decisions as well as those resulting from application of appropriate production rules. The current state of the specification development process is always reflected by the class of specific requirements data and this information, along with the completeness rules provided, allow the user to evaluate his progress, identify unresolved performance issues, and focus his attention on high priority concerns.

The remainder of this section focuses on describing the primary "dimensions" of the knowledge base and the three broad classes of content noted above. It describes the principal dimensions of the knowledge base, then deals with generic requirements data, specific systems data, and production rules, respectively.

Primary Dimensions

The information in the knowledge base will be organized in terms of four primary dimensions to provide a structured framework for the development of system performance requirements. These notional dimensions are: level of resolution, mission category, echelon, and system type.

Level of Resolution. As discussed in the section "Basis for Development," this dimension portions the state space on a temporal and spatial basis. The anticipated levels of resolution are: Theater/Army year, Corps campaign month, Division week, Battalion day, Company/Battery hour, and System minute. This is one of the principal dimensions of the state space discussed in the section "Background," insofar as the series of system states will include residency in most, if not all, of these locations in time and space. We are not proposing to evaluate theater or corps combat performance but to define the potential states of the system for which performance requirements and criteria are required. At the theater level, for example, we are concerned with a structure that will allow the user to: (1) explore the individual system performance requirements implied by consideration of the theater operational perspective, and (2) identify the operational conditions implied by consideration of employment of a system in a particular theater.

Consideration of the theater operational perspective leads to such issues as: deployment to the theater; available facilities at points of embarkation and debarkation; storage and maintenance of war reserve stocks; support and sustainment within the COMMZ; and, for some systems such as Patriot, operational missions in direct support of the theater mission. All of these issues can be derived from consideration of missions peculiar to the theater-level of resolution, some examples of which are shown below:

Type Theater Missions

Mission Category	Design Issues
Movement	Will system be pre-deployed or will it participate in strategic deployment?
	If it will participate in strategic deployment, what modes of deployment are approriate: self-deploy, deploy by air, deploy by sea?
	What size and weight constraints are implied by deployment mode selected?

Type Theater Missions (continued)

How long should it take to prepare system for deployment? What outside assistance (external to system and crew) could/should be available to prepare for embarkation?

Operation

Will system be deployed as part of POMCUS sets, or theater war reserve stocks?

Will storage be land-based or sea-based?

What is desired shelf-life, level of in-storage maintenance requirements?

What are requirements to prepare system for operation: (1) at storage site prior to shipment, and (2) upon arrival at using unit?

Resolution of issues such as those noted above are an integral part of the process of developing system performance requirements and are the result of considering potential missions from the theater perspective. Identification of the particular theater(s) in which it is anticipated that the system will be employed provide information concerning theater-wide operational conditions as well as information that will assist the user in determining the answers to the questions noted above. Theater-wide conditions include terrain, weather, potential employment of chemical and/or nuclear weapons, and the general nature of the threat anticipated.

Consideration of the corps operational perspective will be appropriate for most systems, with the exception of those deployed solely in support of the theater and serviced by COMMZ assets. At this level we are concerned with such issues as: intertheater deployment, interoperability with non-US assets (RSI: rationalization, standardization, and integration); mission performance in direct support (DS) and general support (GS) of corps missions; rear area operations (RAO); support requirements while in the corps rear; communications interface with corps elements; and survival in the corps rear area.

Identification of particular corps in which the system might be expected to be deployed serves less to describe broad ranges of operational conditions than to assist in determining special operations in which the system might participate. For example, potential assignment to the XVIII Airborne Corps suggests special performance considerations that would not be applicable to other US corps.

Consideration of the divisional operational perspective will be appropriate for many systems that will be organic to maneuver divisions or required to provide general or direct support at the division level. At this level we are concerned with such issues as: administrative moves, direct support and general support missions in support of divisional operations, survival in the division rear, support assets available to the division, and potential interface with divisional communications nets: admin/log, operations, fire support, and intelligence.

Identification of the types of divisions with which the system might be associated provides additional information on the types and range of operational missions that may be appropriate. Potential assignment to an armor, mechanized, or motorized division suggests a range of employment options that are not commensurate with employment as part of, or in support of, an infantry division. Similarly, potential assignment to an airborne or air assault division raises a number of performance issues peculiar to those division types.

Since the majority of military systems are deployed under the control of a battalion organization, consideration of the battalion perspective is pertinent to most systems and leads to issues concerning: tactical movement, command and control, and, depending on the type of battalion, a significantly more hostile environment. The potential type of battalion to which a system may be assigned, primarily in terms of combat, combat support, or combat service support provides additional information required to completely describe the potential operating environment. Assignment to a particular battalion type (including the fact that some systems will not be assigned to such an organizational entity) has significant implications both in the areas of communications and reliability, availability, and maintainability (RAM).

At the company/battery level of resolution, the focus is on the organization environment, tactical communications, and participation in company/battery-level missions. At the system level, the focus is on individual system performance, independent missions, operator maintenance, autonomous operation, and internal communication and coordination.

Mission Category. The mission category dimension partitions generic requirements information on a broad mission basis. Based on the idea that the Army has to move, shoot, and communicate, the mission categories anticipated are: movement, operation, communication, survival, and sustainment. These categories provide a functional approach to requirements information and, additionally, map nicely into the categories of capability data typically required by combat models, namely: mobility, performance, command and control, vulnerability, and support requirements.

Echelon. The intent of echelon is to characterize, for a particular system type, the organization level(s) at which the system will be employed. For example, while a close combat system will normally be employed only in direct support at the company level, artillery systems can be employed in direct support of a battalion, in direct support of a division, or a corps level, in general support.

System Type. It is anticipated that the values for system type will be selected to correspond with current Army mission areas, for example: close combat, aviation, artillery fire support, air defense, command and control, and intelligence.

Generic Requirements Data. As noted above, the generic requirements data will be concerned with the types of missions to be considered and the requisite types of conditions and range of derived performance requirements required to specify the mission and to develop the complete and unambiguous performance and RAM criteria implied by the mission.

This information will be organized in terms of the primary dimensions to provide a structured framework for the development of specifications.

The Cartesian product of mission category and level of resolution provide the overall framework for the organization of the generic data and the basis for assessing the contextual completeness of the system specification under development. At the highest level, the generic data will consist of a list of collective missions appropriate to each level of resolution and mission category, for example:

MSNLIST(Level_of_resolution, Mission_category, [list of CMSN])

Collective, or generic missions, are very broad in nature, and are not intended to be directly translated into functions and associated performance requirements. Consider, for example, the collective mission (associated with level of resolution = theater, mission category = movement) of intratheater deployment. The applicability of a particular collective mission to a particular system is defined unambiguously, solely in terms of system type and anticipated echelon(s) of employment, using the conceptual structure shown:

CMSN(Name, [list of SYSTYPE], [list of ECHELON], [list of MISSION])

For the particular example of intratheater deployment, both the [list of SYSTYPE] and [list of ECHELON] would be exhaustive, since we do not currently anticipate development of requirements for systems dedicated solely to CONUS defense. On the other hand, collective missions such as "Tactical road march" or "Provide corps air defense" will only be appropriate for a limited number of systems. The basic definition of completeness will be dealt with at this level and is discussed in the section "Production Rules." Essentially, a complete system description must include all collective missions appropriate to the system type and proposed echelon(s) at which it will be employed.

Collective missions will be described, as noted above, as a list of specific mission templates which will be used to develop sets of conditions and performance requirements appropriate to the system under consideration. A notional mission template would be:

MISSION(Name, [list of functions], [list of conditions])

For the collective mission of intratheater deployment, we would anticipate a number of mission templates:

- Pre-deploy,

- Self-deploy,

- Move to port of embarkation (POE),

- Prepare for embarkation,

- Embark,

- Disembark at port of debarkation (POD), and

- Prepare for movement from POD.

Note that, for a given collective mission, only a subset of the mission templates may be applicable. For example, large systems such as Patriot may be considered solely for pre-deployment, while aviation systems may be considered for self-deployment. A single system may also be a candidate for various deployment missions. It should also be noted that missions are unique to a particular collective mission, the mission of "Move to POE" will be unique within the generic data structure as a component of the collective mission "intratheater deployment." On the other hand, for a given mission, the functions to be performed and the conditions under which they are to be performed will not be unique to a specific mission. The functions appropriate to the mission "Move to POE," (e.g., prepare for road movement, move administratively over improved highways, refuel, etc.) are also appropriate, and will be linked to other missions involving movement, including "Prepare for movement from POD." This not only simplifies the user's task (encountering a previously encountered function will allow the user to review associated requirements and specifications and accept them as is, or with slight modifications) but provides the basis for insuring the consistency of requirements.

Specific System Data

As noted previously, specific system information will be maintained to reflect the current state of specification development. It will consist of descriptive system information and specific requirements data which are created or modified based either on specific user decisions or as the result of the application of a set of production rules. The knowledge base will be structured to allow specific requirements data to be maintained for multiple systems, but, for the sake of this discussion, the system dimension will be ignored.

Information describing the system will be developed as the user responds to various inquiries during the development process. This information is concerned with the type of system, the echelons at which it will be deployed, and the type of support to be provided. As the user addresses various levels of resolution, additional information will be collected in terms of anticipated theaters of employment, the types of divisions to which assignment is anticipated, the system role (in terms of combat, combat support, or combat service support), and other information required to specify the proposed employment and use of the system. A summary of this information will be available for review of high-level employment issues, separate and distinct from the detailed requirements specifications that will be developed.

It is worth noting at this point that one of the dangers of the proposed comprehensive treatment of performance requirements is the apparently inherent bias to include (as opposed to exclude) any criteria whose applicability is actually uncertain. In other words, it is easier, faced with the question as to which theaters a system might be deployed, to select all possible theaters. While a piece of communications gear might be appropriate for deployment with all division types, specifying that a 155mm self-propelled howitzer is to be assigned to an infantry or airborne division implies performance requirements that are completely inappropriate -- and generally expensive to meet. Production rules will be used in an attempt to mitigate this tendency.

The specific system performance/requirements information structure will parallel that used for generic data and specific structures will be instantiated for a system as decisions are made. For example, once the user has specified the system type (SYSTYPE) and echelon (list of ECHELON) for a system, a specific MSNLIST structure will be instantiated with each [list of CMSN] limited to collective missions appropriate to the specification. Note that this will be a reversible procedure; the user will be able to add or delete echelon specifications to the list of ECHELON for a system, or to explore the potential impact of doing so.

Similarly, as the user selects appropriate missions to describe a collective mission, the associated [list of MISSION] will be adjusted. Production rules will insure that the list selected is sufficient to fully describe the performance associated with a particular collective mission. Since completeness is defined in terms of: (1) all collective missions appropriate to system type and echelon, (2) selection, for each collective mission, of a sufficient set of missions, and (3) expression of each mission in terms of necessary and sufficient conditions and performance requirements, the information structure will be augmented to allow the system to track and display the user's progress.

At the requirements and conditions level of specificity, information will include the identification of conditions that must be specified (for example: weather, terrain, combat intensity) and the range of derived performance requirements (for example: relocate, go into operation, provide final protective fires) that provide a more specific, but as yet incomplete, specification for a particular mission. We anticipate that initial requirements/conditions at this level will be selected by the user, with assistance provided by the production rules. As the specificity of the system requirements information increases, increasing information at this level will be automatically completed by production rules.

At the criteria level, the knowledge base will contain the specific, measurable performance and RAM criteria that will ultimately comprise the system specification. Information at the criteria level will also be used to insure consistency of the specification in terms of specific performance requirements. While various missions may independently generate specific requirements (for example: mean time between failure, operational availability, firing rates, and information processing rates) the knowledge base will maintain a single criterion for each performance requirement (subject to condition ranges), linking them as appropriate to information at the requirement and mission level of specificity.

Production Rules

The production rules, in combination with the generic requirements data, provide the expert knowledge required to assist the user in the development of a complete and consistent set of specific system performance requirements. The intent of the rules is to automate, insofar as possible, the development of specific system requirements information at the mission and requirements/conditions level and to provide significant assistance to the user in developing specifications at the criteria level. The rules will have full access to all knowledge base contents, and will solicit additional information or decisions from the user as required.

The production rules will contain, in addition to implicit information and expertise concerning specification development, explicit information that will be provided to the user when soliciting information or a decision. A production rule soliciting a user for a decision will provide:

- information on the choices available;
- optional information on appropriate references to consult for additional information;
- optional information concerning the reasons a decision is required, in effect, the sequence of prior decisions and specifications that led to the current decision point; and
- optional information concerning the implications associated with each of the available choices, including other mission categories and levels of resolution.

When soliciting information, particularly at the criteria level, production rules will provide explicit information to the user concerning:

- information on the range of acceptable data values;
- information on previously specified criteria that are related to the current requirement, including the current specified value for any criterion previously specified with respect to any mission or requirement other than the one currently under consideration;
- optional information concerning the implications associated with each of the available choices, including other mission categories and levels of resolution;
- recommended sources to be consulted in order to develop the required information; and
- if use of a combat model is indicated, details concerning model input requirements (any of the information currently available in the knowledge base will be provided automatically, subject to current specificity of requirements) and resultant measures of effectiveness that are of interest.

As noted, the purpose of the production rules is to assist the user in developing a complete and consistent set of performance specifications, keeping track of the specifications developed to date, and, insofar as possible, avoid over-specification. The:e major functions are briefly discussed in the following paragraphs.

Completeness is defined in terms of: (1) all collective missions appropriate to system type and echelon, (2) selection, for each collective mission, of a sufficient set of missions, and (3) expression of each mission in terms of necessary and sufficient conditions and performance requirements. Thus, the production rules required are not particularly complex, and are primarily bookkeeping functions augmented to provide the user with appropriate status information. The essential elements of completeness are defined by the generic requirements data structure.

Production rules for maintaining consistency of requirements promise to be somewhat more complex. As noted in the discussion of the generic data, functions to be performed (and the associated conditions and performance criteria) are not unique to a particular mission, and a single function or RAM criteria may be encountered in the context of a variety of missions. Once a function is initially defined, it is available for review and modification if subsequently encountered in the context of some other mission. This is the easier component of consistency to deal with: if criteria for cross-country mobility (in terms of speed, range, load limitations, fording capability, etc.) have already been defined in terms of tactical movement mission at the division level of resolution, these criteria merely require review in the context of a relocation mission at the company or system level. Since multiple definitions of specific criteria (fully qualified in terms of conditions of performance) will not be allowed, consistency at this level is guaranteed.

The more difficult component of consistency checking is concerned with related but distinct criteria. For example, without some intelligent controls, it would be possible to independently develop criteria for firing rate and time to replenish on-board ammunition such that the weight or volume of on-board ammunition required would exceed total system weight or size constraints developed in the context of deployment missions. While this example is slightly far-fetched, it is indicative of the requirement to develop a gross performance model to represent the interdependencies of the entire set of performance criteria for a particular system. It is our intent to provide a generic performance model as well as the production rules required to adapt this model to the particular system under consideration.

Production rules to keep track of the specifications developed to date are straightforward; the principal challenge is to structure them in such a way that they provide the user with appropriate measures of progress and a means of estimating the effort required to complete the specification. The prototyping development process proposed will allow us to experiment with these rules to develop a set that will be usable by and useful to the ultimate user.

The challenge of developing production rules to avoid overspecification of requirements is clearly as much of a technical challenge, if not more so, as that of developing a complete and consistent set of performance requirements. While the history of Army systems development contains many examples of seriously incomplete system specifications that lead to requirements for expensive modifications during low-rate initial production or subsequent to fielding in the form of product improvement packages (PIP), it is similarly replete with examples of expensive over-specification. As an example:

- The AN/TSQ-73, Missile Minder, original specifications called for this air defense command and control system to interface with approximately 30 radars (of all three Services), some essentially obsolete. Millions of dollars were spent on developing the required interfaces before it was determined that the actual requirement was the HiPar and LoPar radars being deployed with the Hawk and I-Hawk batteries.

- The unconstrained growth of the performance specifications for the Mechanized Infantry Combat Vehicle (MICV) in the early 70s led to a determination (by OMB and others) that the system could not be built ... this was one of the Army's "Big Five" systems.

RAM criteria in particular, since not well-understood, are prone to over-specification. Not atypically they are developed in terms of an across-the-board improvement with respect to the RAM criteria for the system being replaced without regard for cost, actual performance impact, or the actual (as opposed to specified) RAM performance of the system being replaced.

It is unclear, at this point, precisely how this problem will be dealt with. Unfortunately, while "unambiguous performance criteria" has a nice ring to it, it is difficult to develop unambiguous performance requirements in many cases. While the requirements that a system be air-transportable, or that it communicate via FM, have a significant impact on a system's utility and can be clearly defined and declared not subject to negotiation, range and firing rate are much more ambiguous. By this time, hundreds of analytical person-years have been spent in an attempt to determine the desired range of: the Army Tactical Missile System (ATACMS); its immediate predecessor, the Joint Tactical Missile System (JTACMS); and the JTACMS immediate predecessor, the Corps Support Weapons System (CSWS). The development process has been continuous, despite the name change, and the proposed range requirements have varied from approximately 100 km to approximately 300 km. Clearly, each additional increment of range provides some improvement in operational capability. The principal issue is the magnitude of improvement with re spect to the associated costs for a given increment. The best one can hope for is to provide some type of utility function for range to allow rational design trade-offs. For the particular example of TACMS, the utility of additional range is strongly linked to assumptions concerning the ability to acquire and identify suitable targets as a function of range. This depends on, among other things, the performance criteria for other systems currently being defined and developed.

Another potential approach to the problem of over-specification is to allow the expert system to employ inexact reasoning. Various schemes are being developed employing such techniques as three-valued logic or fuzzy sets to allow a multi-valued logic system to be employed. In such a system, propositions (such as: system will be deployed in support of mechanized divisions) are not either true or false, but may have a value expressing various degrees of uncertainty. EMYCIN (Essential MYCIN), derived from one of the earliest expert systems, is an expert system environment that employs this technique. While this approach is appealing from an academic sense, it is unclear whether it offers sufficient advantages to be chosen as the basis for solving the problem at hand.

Populating the Knowledge Base

The purpose of this section is to discuss the sources of information and expertise that will be incorporated in the knowledge base. The specific structure of the knowledge base and the manner in which information will be incorporated and by whom is still under discussion. While it would appear desirable, on the surface, for users having expertise in appropriate areas to be able to add production rules and generic requirements data to the knowledge base (either explicitly by means of an expert user interface or implicitly via adaptive production rules), the conceptual design has not matured to the point that alternatives to the knowledge engineer have been developed.

We are concerned here with the classes of knowledge base content that can be prespecified: the generic requirements data and the production rules, and each class is discussed separately.

Generic Requirements Data

It should be noted that the generic requirements data comes in two forms: (1) specific missions, functions, tasks, and conditions that will be included as data elements; and (2) the relationships between these data elements that will be reflected primarily in the structure selected for the knowledge base. We are principally concerned here with the first form of generic requirements data. The second form of generic requirements data is contained explicitly in the wide range of combat models and associated military performance evaluation methodologies that have been developed during the past 20 years. Evaluating, estimating, and refining these relationships in order to estimate system performance and its effect on the Army's ability to fulfill its assigned mission reflects the principal technical activity of the VRI project staff that were assigned to this program.

It would be possible to develop the indicated data elements from a combination of first principles and a survey of mission and performance criteria contained in such documents as letters of agreement (LOA), mission element needs statements (MENS), statement of required operational capability (ROC), operational testing (OT) and developmental testing (DT) test designs, force development test and evaluation (FDT&E) test designs, and other sources. However, the requisite data is generally available in significantly more usable form. As an example, the Human Resources Test and Evaluation System (HRTES) provides system missions (categorized by system class) as well as most of the associated function and task templates that would be needed to develop the required data elements for two cells of the conceptual matrix discussed in the preceding section:

(mission category-performance, resolution-company/battery) and (mission category-performance, resolution-system). Similar systems and associated military mission-function-task taxonomies have been developed under ARI auspices that would serve well as data sources. We are not proposing additional studies or exhaustive literature searches to develop data where it is not readily available, since omissions will be detected and resolved during a logical and systematic review of the data, which will be a key step in the development process.

Production Rules

The principal source of information and expertise for the construction rules will be experts, in the areas of military analysis, who are intimately familiar with the development and employment of the combat models and analysis methodologies used by the Army to estimate the effects on combat performance of modifications to doctrine, organization, and equipment, and who are familiar with military operations at all echelons and across the full range of mission areas.

DEVELOPMENT AND DEPLOYMENT OF THE AID

For the aid described in the preceding section to be of benefit to the Army requires several things of the aid. It must be usable within the limits of the resources that the Army devotes to the requirements process, that users be trained in its use, that they accept it and find it useful, and that it be adopted for use within TRADOC development organizations. In addition it must be developed within the resources available. This section describes how those goals will be met. It describes procedures to insure acceptability and usability, describes how the aid will train its users, presents the resources required for the aid's development, discusses the person-hours that would be required to apply the aid to develop requirements for a major acquisition, and describes how the aid can be institutionalized. These topics correspond to items six through ten of the concept paper contents listed in the contract.

Procedures to Assure Acceptability and Usability

Acceptability and usability can be assured by: (a) designing an aid that solves the problem faced by the user and does so in a way that the user will want to work with, and (b) developing the design into software through prototyping in concert with the user, and introducing the aid into the user's work environment during the process of development.

This section has four parts. It identifies the organization and job type of the intended users of the aid, along with the phase of the acquisition cycle when the aid will be used, explains how the development approach selected will assure the acceptability and usability of the aid to the intended population of users, describes how the software developers and users will interact during development to assure acceptability and usability, and describes the features of the aid that the detailed design will have to possess to guarantee this goal.

User Identification

To guarantee an acceptable and usable aid it is necessary that we design and develop it to suit the persons who will be using it. We must know at least the following:

(1) organization,

(2) job type, and

(3) acquisition phase when the aid will be used.

Requirements generation is the responsibility of junior officers in the combat development organizations of TRADOC schools and centers. It occurs during the concept development phase of the system development cycle. These persons work within an environment filled with meetings and deadlines, in which time for doing productive work is limited. They often have not been at the job long and usually have little familiarity with the requirements process. Often they lack experience with other areas of the Army outside their own specialization, as well as with threat elements that can impose requirements. As a result, they will often be lacking in the breadtn of experience needed to identify potential dimensions of performance where requirements are needed, as well as lacking in the depth of experience needed to set quantitative criteria for required performance -- without some guidance. The computer skills in this population will be quite varied, as well as attitudes toward the use of the computer. Familiarity with combat models will not be common, although there will be formal liaison and some form of cooperation with groups of combat modelers (although perhaps not in a dedicated role during the requirements phase). These people will likely have had some role in a preceding mission area analysis in which the need for the system was identified and justified.

Decision Support Development Approaches

Insuring the acceptability and usability of the requirements generation aid is principally a development issue. That is, the aid must both be designed with user acceptance and usability in mind, and the design must be modified during development as prototype versions are tried by typical users. Problems of acceptability and usability are common throughout the field of software development, and general approaches have evolved within the software engineering community to solving these problems. For that reason it is necessary to discuss alternatives for software development approaches to decision support systems.

The development of software for automated computing systems has been and continues to be a problem for the Army, so much so that a two-week Army Science Board Study was conducted on the topic in 1983.[2] In the following paragraphs we summarize alternative approaches to developing

[2] Yaru, Nicholas, "Acquiring Army Software," Army Science Board, 1983.

software for Army computing systems. In discussing these approaches, it is reasonable to consider two types of computing systems: prespecified and user-driven.

Prespecified computing systems are ones in which requirements can be specified in detail and remain stable during the life of the system. An example is a missile guidance computer which implements control algorithms for guiding the flight of the missile to the target. For such systems, conventional development techniques are usually applied. These techniques usually involve five general activities as noted below:

1. Perform r^quirements analysis: In the missile guidance computer example, this would entail determining the kinds of targets the missile would engage, the target flight characteristics, environmental conditions, the input tracking information, and the control relationships. These would be used to formulate the missile flight requirements.

2. Prepare detailed specifications for the code to implement the requirements identified in step 1.

3. Obtain user approval (requiring fairly detailed user review of the specifications).

4. Design, program, and test the code.

5. Prepare system documentation (technical results, maintenance manuals, etc.).

Although this conventional development approach for prespecified systems has been relatively successful, there remain significant problems in the Army's development of software even for these types of systems where requirements can be specified a priori. The development of software for prespecified systems was the focus of the Army Science Board inquiry.

In contrast to the prespecified system, in user-driven automated computing systems the end user cannot specify, a priori, information and analysis requirements in fine detail. These systems dynamically change over time as experience with the system is gained. Accordingly, as substantiated by extensive experience in industry and government, conventional development techniques do not work for user-driven systems. Experience further suggests that when conventional development approaches are attempted, they increase development costs by one to two orders of magnitude, increase the development time by approximately a factor of five, and produce systems that are not responsive to succeeding managers whose information requirements will certainly vary in substance and form from their predecessors.

The prototyping approach to developing decision support systems, such as the requirements aid, does not abandon the conventional phases of development (requirements analysis, specifications, design, implementation, and test), but supports an incremental approach to providing full system functionality, allowing some increments to be fully tested and deployed while others are still i, *he design phase and still others exist only in terms of a vague requirements statement. The principal utility of the prototyping approach, whether used to develop planned increments within the context of an overall design or to define and develop applications in response to newly emerging requirements, is the ability to implement quickly a prototype solution which can be used for three purposes:

1. to provide a timely and effective, if not efficient, response to the stated user requirement;

2. to allow the user actually to experiment with the prototype and refine his requirements to reflect the capabilities he actually needs and wants, and is capable of using; and

3. to allow the system developer and user, working in concert, to develop a truly user-driven and understood requirements specification for a particular application before proceeding with the final development process (which is concerned with developing, testing, and deploying an efficient implementation and adding it to a general user repertoire of capabilities).

Thus, the prototyping approach provides early operational capability while significantly reducing the time required to develop the formal capability specification -- the refined prototype, approved by the user based on actual use, becomes the specification. In a very general sense, the users' manual becomes a significant design document.

The function of prototyping is to allow incremental growth and refinement of system capabilities in response to the evolving requirements of current requirements analysts, not in response to a set of prespecified requirements arrived at by the designer alone. While the designer of the aid, VRI, feels that it understands the kinds of considerations that should enter into requirements generation, and while it feels that this understanding is a unique contribution that it would bring to the design and de'elopment of the aid, it nevertheless realizes that the aid must meet the needs of the user within his working environment, as the user himself perceives his needs. By using prototyping as its development approach, it can be assured that the aid combines both VRI's understanding of the Army's problem and the user's perspective of what he needs to perform his job. Experience has shown that combining iterative, experimental, hands-on experience by the user with responsive modification and enrichment by the system developers will produce software that is useful and usable.

Interactions between User and Developer

In the preceding discussion of the prototyping approach to software development there was an emphasis on the importance of cooperation between the developers and representative users during development. By actively involving users in the development phase the prototyping approach serves to guarantee that: (a) the aid performs functions that they need and it performs these functions in ways that the users can work with, and (b) future user acceptance is enhanced by actively involving them in the development process (i.e., within the TRADOC community it will come to be seen as "their" tool). In each of the three phases of development through which the aid will progress there will be different kinds of cooperation between developers and users. User involvement in each phase is described below.

Detailed Design Specification Phase. During this phase it will be necessary to contact requirements personnel, inform them of the purpose of the aid, and obtain their participation in the development process. Once participants have been obtained, their first contribution to the conceptual phase will be to participate in discussions with the development team about the requirements generation process and their roles within that process. These discussions will lead the development team to refine its understanding of the users' needs for assistance, of the kinds of information they might find useful, and of the environment within which they work. The development team at this time would explain their understanding of the difficulties that exist within the requirements process, including not only problems that are faced by the users but also problems that the Army faces because of the failure to develop complete, consistent, and realistic requirements. At this time the development team will explain in more detail the concept for the aid, the ways that it would help the user, and how it could be used within the context of his day-to-day work. Comments from the users will be considered in refining the concept and in the next phase, detailed design specification. We anticipate selecting a TRADOC school to visit, beginning in this phase of development, where we could place our discussions within the context of a system for which there is current interest in requirements development. Basing our discussions on an existing problem of interest to the users will be the best way to get their interest and participation. It will also facilitate working at a realistic level of detail and specificity to assure that we understand exactly how the user operates, and so that the user understands the intended operation of the aid. At this time the details of design can be altered to accommodate new insights into the skills of the users, the information that they have available, and the outputs of the aid that would be helpful to them. At this time we will be able to revise the detailed design if the user finds that it does not provide the support that he thought it would provide.

Product Production Phase. In this phase the user will be able to work with versions of the aid, beginning with limited prototypes and continuing with complete versions of the system. By this phase the general function of the aid and the user's overall method of using it will have been established. We are likely at this time to have users work through requirements problems with the aid, observe their use of the aid, and discuss their experiences with them. Issues likely to be addressed in this phase will probably concern such aspects as the details of its interaction with the user (formats of screens, e.g.), completeness of areas of the knowledge base, support in linking requirements to analysis tools, and other details of the aid's operation. Basically, as users will gain "hands-on" experience with the tool, they will be able to tell us what they like and do not like about it, and we will be able to revise the software accordingly. This process will be possible because of the use of the prototyping approach to software development, in which versions of the aid are made available to the user early in development. As the user gains experience with the tool, he will be able to see the ways that the tool allow him to go beyond the techniques previously available to him, and the explanatory facilities of the tool will let him see the value of these improvements in the completeness and consistency of the requirements that he generates. In this way the user will come to value the tool, as the tool is adapted to his needs. User acceptance and usability of the aid are thus both addressed by starting with a design that addresses the user's real problems and by developing through prototyping.

Design Features

One aspect of our approach that helps insure the acceptability and usability of the aid is to include in our design features that make the tool easy to use and that endow it with capabilities that the user wants. We have already discussed how our design will assure that the tool will address the right questions and will provide the user with help that he needs. We must in addition design the tool so that he will want to use it. Design features of this sort are:

1. Turnaround time. This must not be so long as to be incompatible with the constraints places on the users.

2. Adequate output. The aid must allow the user to look at his results on the screen or in printed copy, it must allow him to view portions of inputs and outputs according to options under his control, it must allow him access to ongoing and old problems, and it must assist him in comparing the results of different problems (e.g., different solutions for the same system or solutions for similar systems). The output must include the ability to see the knowledge base logic that led the aid to certain solutions.

3. Audit trail. The aid must allow the user to see: (a) what inputs he has provided and what outputs the aid has given, (b) the order in which transactions between user and aid have occurred, and (c) the steps by which the aid has arrived at its responses. It must also allow backtracking and revision.

4. Archiving. Problems must be saved in various stages of completion and retrieved later for work or inspection. The aid must protect incomplete work against loss or destruction.

5. Ease of use. The aid should not require great memory, refer to unfamiliar terminology, be overly complex, or otherwise discourage use. It should contain "help" features to guide novices and to help in the use of infrequently used features.

6. Training. The aid should educate the user in its use.

7. Understandability. By arrangement and design of menus, by logical flow of operation, by explanation to the user, and by other techniques the operation of the aid should be clear to the user, and it should help the user at all times to understand where he

is within the overall problem and where he ought to be going in the next steps.

8. Power and Flexibility. The aid should not burden the experienced user with too many steps that only novices would need to perform. It should offer enough features that users can adapt it to their problems and not be forced into standard solutions.

In addition to the above characteristics relating to the user's operation of tne aid, other features need to be available in the background to allow for the maintenance of the knowledge base by system developers. The knowledge base should be easily extensible and should have an on-line data dictionary for reference purposes. It may be necessary in the future to add entirely new sections to the knowledge base to apply to new classes of systems or to make distinctions among systems aggregated by the original knowledge base into a single class. Separate classes of systems will require separate sets of rules and data in the knowledge base, and the aid should support the extension of the knowledge base to accept new sets of rules and data for new problems. The knowledge base should also allow for modifications within the set of rules and data that apply to a single class of systems. For example, it should facilitate the addition of data for a new theater of operations and the modification and replacement of rules governing relationships among the elements of the friendly force and between them and the enemy force. The aid should also provide for the development of construction rules to automatically resolve data inconsistencies and voids at the time that information is entered into the knowledge base.

How the Aid Will Train Its Users

The aid will not be used unless it is simple to operate and fits conveniently into the work habits of users who are, after all, busy and pressured for time. If the aid requires a concentrated period of training and does not accommodate the occasional user, it will not be used. Consequently, our concept for the aid emphasizes transparent and simple operation of the aid, along with the tailoring of its inputs and outputs to fit the working environment of the requirements analyst. Correct design will result in minimizing the need for training in the operation of the tool.

At the same time we recognize that the need for training is not limited simply to the acquisition of skills to operate a piece of software. The aid is to assist the analyst in understanding the problem of generating requirements. The main need for training is therefore in helping the analyst understand the requirements generation process and the implications of requirements for fielded Army systems. This understanding begins by letting the analyst know how the tool fits into the general class of problems faced by the analyst, and it continues when any problem is being solved by explaining the relevance of the specific requirements and supporting information that it generates. Therefore, there is a significant need for training users in understanding systems requirements prior to beginning an actual requirements analysis. If the aid is easy to use, however, this training need can be met by using the aid on a prepared problem, perhaps for a system of a similar type (e.g., another armored combat vehicle, another artillery system, or another air defense weapon). Since the aid is to assist in understanding requirements, the need for separate training is diminished.

During the development of the aid there are significant benefits to be obtained from reducing the amount of separate training functions to be designed and programmed into the aid. Software for training can be complex and can consume valuable design and programming resources. The development of the courseware to execute on the training software can be resource consuming as well. These resources can be conserved and redirected to improve the aid itself if the need for training is minimized.

The kind of training that we consider to be most important is the review of solved problems, accompanied by explanations of the derived requirements and the steps that led to their derivation. The effectiveness of this training can be enhanced if it is conducted through the normal operating procedures of the aid, with the user following the usual steps of operation. This mode of operation of the aid could be performed with differing degrees of participation by the trainee, ranging from passive following of paced presentation of the sample problem to the active emulation of a full session, in which the aid solicits the user's responses as it normally would in full operation, but filters the responses and guides the solution of the problem along the lines of a correct solution. Whatever the degree of involvement by the trainee, the review of sample problems would further the goals of accustoming users to the aid, demonstrating its value to their jobs, and enhancing their understanding of the problem of requirements generation.

By making the aid simple to operate and by providing sample problems for novices to follow we will have built a tool that would be accessible to a range of users outside the specifically targeted population of current requirements analysts. Copies of the aid could be made available to officer advanced courses, for example, where there is currently little instruction on the requirements process beyond overviews of the system development cycle, major milestones, and documents. The more widespread the use of the aid, the easier will be its acceptance and institutionalization. Ease of use and of training will contribute to these goals.

By following a development strategy of prototyping, we will have the opportunity of assessing the experiences of users with early versions of the aid, and we will be able to follow up with design changes -- which in some cases could be changes to the aid's training facilities. While it would be preferable to amend the design of the aid to overcome any user problems that may be discovered during prototyping, that solution may not always be practical. In that case, it would be necessary to consider alternatives to expand the aid's interactive help facilities to assist users over the problem, or to add to the training facilities built into the aid.

The least desirable solution to such a problem would be the addition of new kinds of training to the aid, specifically, training that is separated from the normal operation of the aid. Separate tutorials or lessons to train parts of tasks would be least involving kinds of training, and an aid that required a great amount of such training would have greater problems with user acceptance. Because we want to minimize the amount of tutorial and part-task training, our detailed design for the aid will not include such training mechanisms. They will be added only as required during prototyping, and our design will incorporate training entirely through playback demonstrations and through sessions that emulate and explain the normal operation of the aid.

Development Resources

This section presents the time, effort, and funds that will be needed to produce the aid. It also estimates the probability of completion by the end of the scheduled period.

The development of an aid which implements the concept described in this paper hinges on the availability of personnel who are subject matter experts in the requirements process. These experts constitute a source of knowledge related to:

1. the requirements process in its own right,

2. the implications of and relationships between performance criteria and levels of performance, and

3. the nature and role of combat models and analysis supporting an iterative determination of consistent and complete system performance requirements.

One approach to produce the aid is to proceed as originally planned. During the next phase (Detailed Design Specification), research staff would specify:

1. required inputs, sources, and access;

2. components, sources, and interactions;

3. processes which produce outputs;

4. outputs, including interfaces;

5. security procedures;

6. means to ensure organizational acceptance; and

7. schedule and budget.

The final phase would utilize the detailed specifications to implement the aid, including:

1. required operational hardware,

2. operational software,

3. required data,

4. training software, and

5. maintenance documentation.

Given the evolution of our concept into one of knowledge engineering, the preparation of detailed design specifications would involve using the concepts of the DePuy matrix and echelon/time focus to specify types of rules, to identify data elements, and to structure a knowledge base. The man/machine interface would be specified and security procedures chosen. An expert system shell and host computer would be recommended. The original schedule for these activities remains appropriate; i.e., eight calendar months, of which six are devoted to the necessary research. Estimates for implementation involve the expenditure of approximately 5,900 person-hours over a period of roughly 24 months.

As noted earlier, a prototyping approach is recommended. As an option to the original plan, an expert system shell could be put in place in the next phase of the research and knowledge engineering conducted in the context of providing support to the specification of requirements for an ongoing acquisition. In this case, however, the schedule might have to be altered and phases two and three integrated into a single implementation phase. Barring the adoption of this approach, it is probable that the team would use a specific system to focus its initial efforts in the detailed design phase, expanding to encompass representative systems from the remainder of the major battlefield subsystems.

Application to a Major Acquisition

This section concerns the number of person-hours that will be required to use the aid in developing requirements for a system that will be a major acquisition.

Because this aid is intended to act as an expert advisor and guide during the solution of a requirements problem, it is envisioned as a tool that analysts will consult on a continuous basis during requirements generation. What requirements analysts need is guidance throughout the process, which is a situation different from a single, isolated problem for which the analyst would turn to the aid once or twice to ask for an answer. Under this kind of use the goal is to provide an aid that analysts will use frequently and will integrate into their work habits, rather than to provide a tool for one-time use. If the latter were the case, it would be necessary to minimize the amount of time devoted to the aid's use, so as to make it acceptable to the user. We want a different kind of aid, however.

Nevertheless, only a limited amount of time is available to the user, and it is important that the aid be unobtrusive and not lengthen the analyst's job. Hence, we want to know the aid's impact on the total time needed to generate requirements, as well as the time required for the use of the aid itself.

In determining the impact of the aid on the total person-hours needed for a major acquisition it is necessary to know a baseline number of person-hours required for a problem without the use of the aid and to know the number of person-hours required for the same problem with the use of the aid. The difference between the two is the impact of the aid on the number of person-hours required for solution of the requirements problem.

As noted earlier in this paper, the preparation of a requirements document is a lengthy process in which multiple drafts are generated and revised. One estimate of the magnitude of combat developer resources expended for a major system acquisition was approximately 10,000 person-hours over two or more years. Analytic support was estimated to require in excess of 10 person-years. The concept proposed in this paper is intended to support the combat developer. As such it will hopefully contribute to reducing the number of times it is necessary to revise requirements by ensuring that the levels of performance are demonstrably consistent and complete. The aid should also contribute to increased efficiency by making available to its users, in a timely fashion, expertise and data that they currently have to search out. As for the amount of time required for the use of the aid itself, we anticipate that multiple users might spend on the order of 200 to 300 hours on-line in the course of a major acquisition. Compared to the total effort involved this amount is acceptable particularly if the aid is used in archiving, process management, and documentation roles. This is only a very rough estimate. A better estimate will be possible when prototype versions of the knowledge base have been prepared and assessed. The time required to use the aid will depend on the nature of the system for which the requirements are being developed and on the extent to which the user of the aid follows up the opportunities offered by the aid. Systems will differ with respect to the quantity of requirements that should be specified for each and with respect to the difficulty of setting those requirements. Also, a user may choose to respond to the aid with partial answers, so as to avoid investigating certain kinds of requirements. Purposeful oversimplification could result in incomplete requirements, but in some cases a user might find it a very practical strategy for avoiding repetition of his effort when two sets of operating conditions differ only in a few respects. The aid can offer that kind of flexibility.

Institutionalization

This section discusses the institutionalization of the aid within the Army. Our approach to this problem is based on the belief that a tool will be adopted if the intended users are given "hands-on" experience with it, find it usable and convenient within their work environment, and see that it helps them with their problems in the requirements process. If the potential users develop this attitude and the aid is placed at their disposal, then they will use it and it will de facto become part of the institutional process of requirements generation. If it produces results, more formal institutionalization will occur, but it would probably not be effective to attempt to impose the use of a tool from above: the attempt would be futile without demonstrated contributions to management of the tool's enhancement of the requirements process, and an imposed solution would not foster actual use by users at the bench level.

Earlier in this section we discussed steps to assure that the aid will be acceptable and usable. These qualities are guaranteed by our prototyping design and development approach, in which the targeted users are exposed to the tool early in detailed design and during development. This approach is also the key to achieving institutionalization.

Prototyping of the aid should be done on a real requirements problem in order to get the full commitment of the participating users. Given the demands on the time of the users, their participation can be gained only if the prototyping is seen as helping immediately with their problems. In addition, working with a real problem will guarantee that the resulting aid is designed to handle real problems under real working conditions, and not artificially contrived or simplified problems. Demonstration of a usable and useful tool to actual requirements developers on a real problem and under real working conditions will not only integrate the aid into their own work environment, but it will also make an effective demonstration for the rest of the Army development community.

Once the aid has gone through initial development, and once knowledge bases and sample problems have been prepared, exposure to a wider set of users would be useful. As discussed in the earlier section on usability and acceptability, at this stage the aid could be provided to officer advanced courses in all the schools, where it could be used to teach students about the requirements process. This is one case where versions of the knowledge base would have to be tailored to an instructional problem, because of the limited time devoted to teaching this topic in advanced courses. In fact, the system development cycle as a whole is treated but briefly in the advanced courses. Versions of the knowledge base developed for full requirements problems would require too much time of these students. However, even if realistic but cut-down problems are solved, and if this process is seen as furthering the students' understanding of the development cycle, the aid will benefit from a good reputation, as well as from becoming known among a wider group in the Army.

In summary, there are two actions that can foster the institutionalization of the requirements aid:

a. to use it at a combat developments center on a real requirements problem, and

b. to make the aid available for teaching sample problems about the requirements process in the advanced courses.

The Army recognizes that there are problems with the level of training of the combat developers. Demonstration of a solution to those problems would be a convincing argument to the Army for institutionalization of the aid. The tool will provide some of the expertise known to be missing in the requirements personnel, and it would allow them to produce improved requirements as they gained valuable training and improved their skills by working with the aid.

MANPRINT METHODS MONOGRAPH:
AIDING THE DEVELOPMENT OF MANNED SYSTEMS PERFORMANCE CRITERIA

PROCESSOR FOR THE DEVELOPMENT OF SYSTEM REQUIREMENTS

Joseph Conroy III, Ph.D.

Science Applications
International Corporation
1710 Goodridge Drive
McLean, VA 22102

Edward Connelly

Performance Measurement
Associates, Inc. (PMA)
1625 Autumnwood Drive
Reston, VA 22094

CONTENTS

CONTENTS (Continued)

Page

LIST OF FIGURES

CONTENTS (Continued)

PROCESSOR FOR THE DEVELOPMENT OF SYSTEM REQUIREMENTS

INTRODUCTION

Requirement

This is a concept paper for product one of the contract entitled "Concepts on MPT Estimation (Development of MANPRINT Methods)" (contract no. MDA 903-86-C-0414). Product one was solicited to help the combat developer generate objectives and performance criteria for new Army systems, and identify conditions that will affect the performance of a new system.

The Army's current approach to identifying objectives, performance requirements and criteria for new systems lacks sufficient reliability, precision and assistance for the combat developer. Often the Army's approach results in new system objectives and criteria that are not related to required system performance. Instead, the objectives and criteria are frequently similar to hardware specifications. Also, the criteria that are related to system performance are not always specifically designed to ameliorate the deficiency that spawned them. (See Appendix A for a thorough description of the need for a system requirements methodology.) Thus the Army needs reliable, user oriented tools for developing objectives and criteria for new systems and identifying the conditions that affect system performance.

Description of the Product

This paper proposes as product one a computer processor that provides a reliable, systematic and precise method of developing requirements for new systems. The processor is to be used by combat developers just after the production of a Mission Area Analysis or any other documentation of a deficiency in the Army's ability to carry out its missions.

The processor develops objectives for new systems and criteria by which to evaluate the performance of systems. In addition, the processor identifies the tactical, operational and environmental conditions which could affect the performance of systems.

The Development of Objectives for Systems

The processor first develops objectives for systems by presenting a subset of its data base of functions, subfunctions and activities to the user in a menu format. The subset presented depends on the type of system being developed. The processor prompts the user to select from the appropriate limited set of functions and subfunctions, those the MAA or deficiency indicates the system should perform. Then the processor formulates an objective from the functions and subfunctions in the form of the following example:

Mission Area = **FIRE SUPPORT**
Function = **TARGET SERVICING**
Subfunction = **SUPPORT/SUSTAINMENT**
Activity = **MOVE**
CARGO
PERSONNEL
CROSS COUNTRY
(etc. for the other specs.)

The performance objectives will be drawn from a data base resident in the processor. The data will be obtained from one or more of several candidate sets. One of these was developed by the TRADOC schools for performing mission area analyses. Another candidate set of functions and subfunctions was developed by SAIC for an ARI project for the purpose of determining the functions and subfunctions of units. Other candidates are available. The most appropriate in terms of user acceptance and ability to provide a comprehensive and accurate set of options will be selected. Then it will be modified as necessary and incorporated into the processor.

The Development of Performance Criteria for Systems

The output of the product one processor includes criteria for judging the performance effectiveness of systems. The processor synthesizes performance criteria from the cognitive decision rules of experts which the experts use to judge the performance effectiveness of systems.

A new system should be judged by its contribution to the effectiveness of its unit (e.g., company) (Campbell et al., 1970; Guion, 1961; Kendal, 1956). Thus the processor determines the cognitive rules of experts by first querying them for their judgements of the expected effectiveness of the system's unit given the system is fielded by its unit.

Then multiple regression is used to identify "predictors" of unit effectiveness. The potential "predictors" examined are all candidate system performance characteristics (e.g., speed or carrying capacity of the system). Those candidate system characteristics the regression analysis finds to be significantly related to unit effectiveness are adopted as the performance characteristics (criteria) the system must meet. The whole process is shown in Figure 1.

Theoretically there are hundreds of candidate system performance characteristics that could be adopted as the performance criteria that a new system must meet. The problem is to identify those characteristics that, if met, would result in the system enhancing the effectiveness of its unit. The processor's approach to this is to initially include many candidate characteristics and then have the regression analysis identify the characteristics that are significantly related to unit effectiveness. Since the judgements of unit effectiveness are given by experts in relation to several example values for the candidate system characteristics, the resulting regression equation embodies the rules the experts use to relate system characteristics to unit effectiveness. Such a regression based

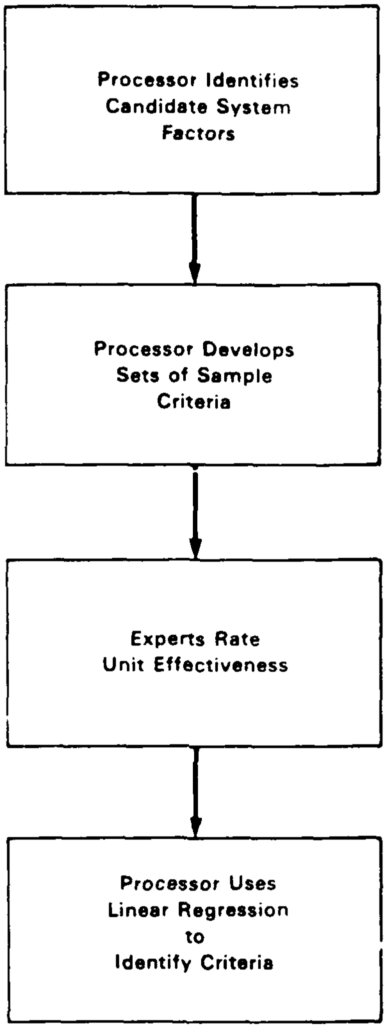


Figure 1 Process of developing system performance criteria.

approach has been used for many years to model the cognitive rules of experts (Darlington, 1968; Dawes and Corrigan, 1974).

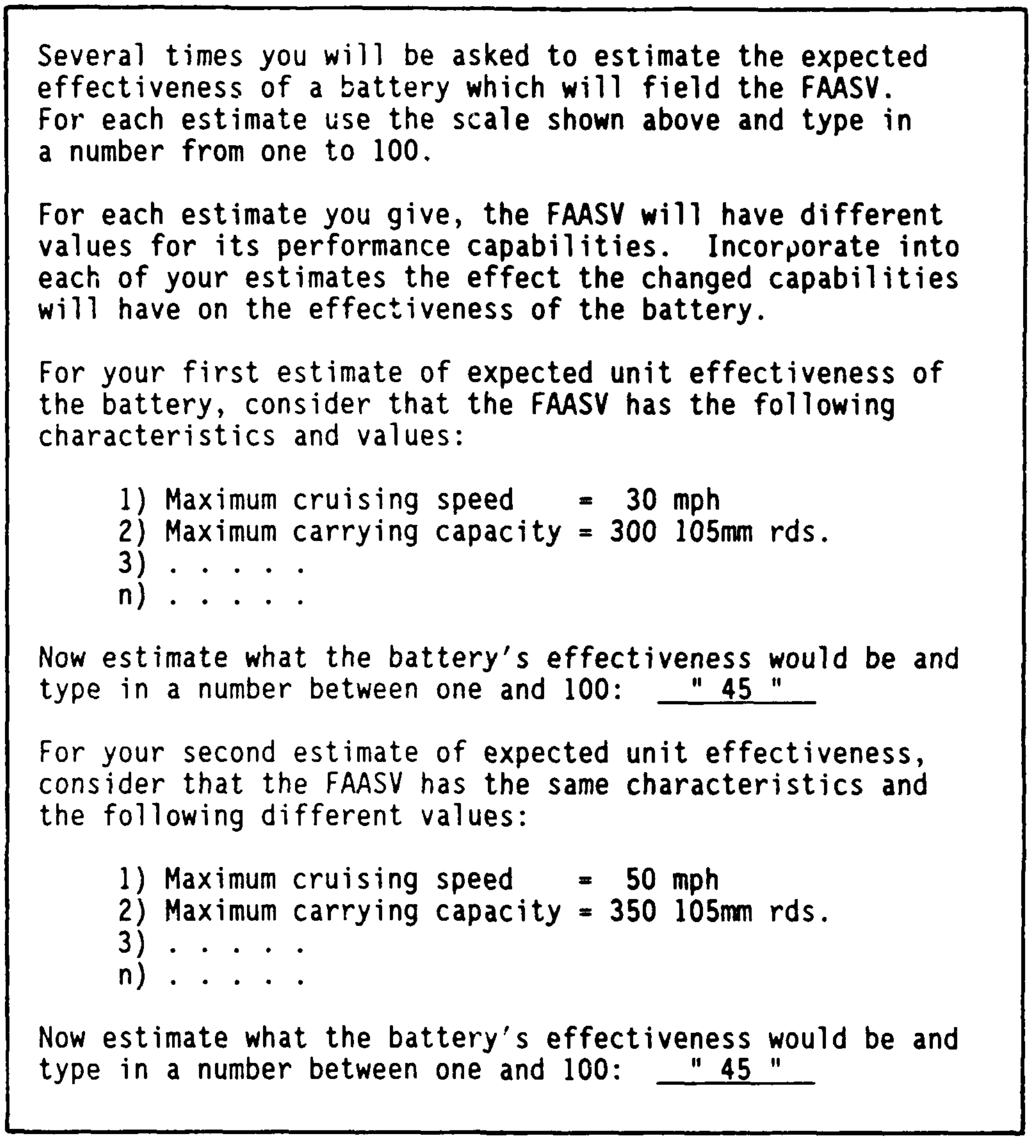
In judging the expected effectiveness of a unit, the experts are shown the MOE for the unit. Knowing the factors and algorithm for judging the effectiveness of the unit will help the experts make accurate judgements (Connelly, 1981). The unit MOE will be produced by the processor's designers during its development and stored in one of the processor's data bases.

The processor obtains the effectiveness data by querying each expert for several judgements of the expected effectiveness of a system's unit. The experts make their judgements of expected unit effectiveness using an "anchored" scale of one to 100; 100 being the best performance. Before each query, the processor changes the values of the candidate system performance characteristics. For example, the value for maximum speed might be changed from 30 mph to 50 mph and the carrying capacity of the system from 300 rounds to 350 rounds so that each expert judgement is based on the new values for the candidate system performance characteristics.

An example of this process as it might occur on a screen is depicted in Figure 2. The figure depicts the process at the point where the processor is querying an expert. Although the system used in the example has already been developed, it provides the basis for a realistic example. The system is the forward area ammunition support vehicle (FAASV). The expert will already have been shown the unit's MOE, deficiency, and examples of an Op order, description of the enemy, mission location, etc.

The processor obtains candidate system performance characteristics from its data base of the characteristics of predecessor and similar systems. These are added to and modified by the user and the experts as they see fit. For each judgement by the experts, the values for the candidate system characteristics are varied within a range by assigning randomly selected values to each candidate system characteristic. The range is defined by the theoretically largest and smallest values possible for each characteristic. The theoretical values are estimated by the user.

After the judgements from the experts are obtained, multiple regression is used to identify the set of system characteristics which are significantly related to the unit effectiveness judgement scores. The multiple regression determines the relationship between the values of the candidate system characteristics (e.g., 30 mph for speed) and the judgement scores (1 to 100) given by the experts. Those system characteristics found to be significantly correlated with the effectiveness of the system's unit are selected by the processor as performance criteria for the system. The selected criteria will have been shown to significantly affect that which the system is to enhance - t'.e effectiveness of its unit. Of course the final choices of criteria are left to the discretion of the combat developers.



Several times you will be asked to estimate the expected effectiveness of a battery which will field the FAASV. For each estimate use the scale shown above and type in a number from one to 100.

For each estimate you give, the FAASV will have different values for its performance capabilities. Incorporate into each of your estimates the effect the changed capabilities will have on the effectiveness of the battery.

For your first estimate of expected unit effectiveness of the battery, consider that the FAASV has the following characteristics and values:

1) Maximum cruising speed = 30 mph
2) Maximum carrying capacity = 300 105mm rds.
3) . . . . .
n) . . . . .

Now estimate what the battery's effectiveness would be and type in a number between one and 100: " 45 "

For your second estimate of expected unit effectiveness, consider that the FAASV has the same characteristics and the following different values:

1) Maximum cruising speed = 50 mph
2) Maximum carrying capacity = 350 105mm rds.
3) . . . . .
n) . . . . .

Now estimate what the battery's effectiveness would be and type in a number between one and 100: " 45 "

Figure 2. Example of obtaining an expert'c ..ents of unit effectiveness.

The output of the processor will include the regression equation and the system criteria including their importances and minimum acceptable scores. An example of the system criteria output might be the following:

SYSTEM CRITERIA

System Characteristics	Importance	Minimum Score
1) Cruising speed	.8	40 mph
2) Carrying capacity	.6	350 105mm rds.
3)	.X	. . .
n)	.X	. . .

These data will be explained to the user and he will be told how to interpret and use them.

The Identification of Operational, Tactical and Environmental Conditions

The processor also will identify the operational, tactical and environmental conditions which may affect system performance. The process for identifying conditions is shown in Figure 3. The approach to identifying conditions amounts to having the processor ask the experts for additional judgements of unit effectiveness at the same time that the processor asks the experts for their first set of judgements. However, when asked for the second set of judgements, the unit's situation will be modified by a candidate list of potentially significant conditions. If the experts give judgements of unit effectiveness which are significantly lower than their first set of judgements, then the conditions will have been shown to significantly affect unit performance. Differences produced by the conditions will be tested for significance with F tests.

The approach to developing conditions will begin with the processor presenting to the user a candidate list of potentially significant conditions. These will be selected from its data base of conditions using a key word search. Key words will be the types of systems whose requirements are being developed. Type of system refers to generic type such as helicopter, tank, howitzer, etc. A conditions data base will be made part of the processor and each condition will have one or more key words attached to them. The conditions will be those currently believed to be significantly related to systems. They will be obtained from extant O&O plans and other requirements documents such as required operational capabilities (ROC) documents. The user can modify and add to the candidate list of conditions.

After possibly being modified by the user, the candidate list of conditions will be presented to the experts when they are making their judgements of expected unit effectiveness. The experts will be asked to make the same type of judgements of unit effectiveness they did earlier for the identification of system criteria. However, this time the experts will be asked to make their judgements of the unit as if it were operating under one of the conditions on the list of candidates. If the experts' judgements indicate that the unit will be significantly less effective than it would be without the conditions, then the conditions will be deemed to have significant effects on the system's performance. If the experts' judgements of the effectiveness of the unit indicate that the unit will not be significantly less effective, then the conditions will be deemed to have no potential to effect system performance.

The design of the processor is based on a simplified version of the mission analysis processor (MAP) of Connelly (1986; 1981) which is currently being used by the General Services Administration (GSA). The MAP processor functions as an aid for developing MOE for units. Many new functions have been added to the MAP processor to produce the new Product 1 processor. The new processor automatically performs most of the functions the MAP processor required the user to perform such as querying experts for

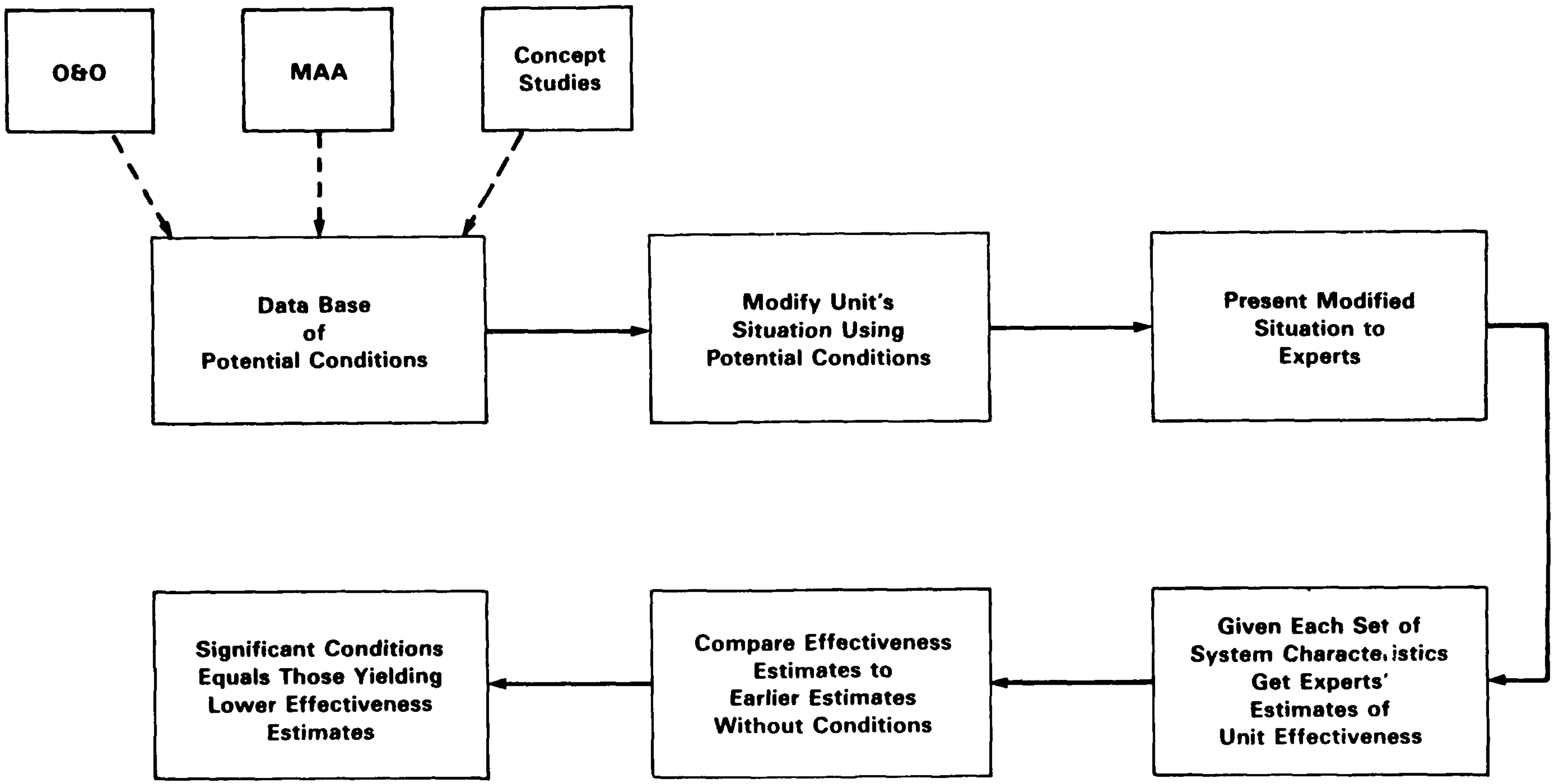

O&O
MAA
Concept Studies
Data Base of Potential Conditions
Modify Unit's Situation Using Potential Conditions
Present Modified Situation to Experts
Given Each Set of System Characte.istics Get Experts' Estimates of Unit Effectiveness
Compare Effectiveness Estimates to Earlier Estimates Without Conditions
Significant Conditions Equals Those Yielding Lower Effectiveness Estimates

Figure 3. Conditions identification process.

judgements. Even though the MAP processor has many new automatic features, it was greatly simplified by removing unneeded portions which dealt with higher order system dynamics such as aircraft flight control.

The next section of the paper describes the functions th^ processor will perform. Following those, a detailed description of how the processor will perform each of its functions is presented.

FUNCTIONS PERFORMED BY THE PROCESSOR

The processor performs three major functions: the development of objectives; system criteria; and the identification of conditions that might effect system effectiveness. Each of these functions is described in turn on the following pages.

The processor will be accessed with the appropriate user name and password or by using floppy disks. The processor will be used by many geographically distant combat developers. Also each user will need to have the processor access experts. Thus the preferred mode of communication between users, the processor and the experts will be electronic rather than shipping questionnaires or disks back and forth. For the sake of simplicity, the following descriptions of the processor and the functions it will perform will assume that both users and experts will have interactive access to a main-frame resident processor via a keyboard and a CRT.

After the user accesses the processor to initiate the development of requirements for a system, the processor will ask the user for the name of the system for which to develop requirements. Upon subsequently accessing the processor, it will ask the user if he wishes to continue where he left off during his last interaction.

A help system, callable at any time, will be part of the processor. It will provide information about the current process and a menu to call information about any other part of the process. Further, a menu system indicating how to change the system from the present state to any other state will always be shown on the bottom line of the screen. Finally, a tutorial will be available to illustrate how to use each of the application aids available with the processor.

Development of Objectives for Systems

The processor will develop objectives for a new system by relying primarily on a data base consisting of several levels of functions and menus for specifying the who, what and where of the functions. The process of developing objectives will begin after the user initially accesses the system. It begins with the processor asking the user for the name of the system and to indicate from a list of units which is the type of unit in which the system will be fielded. Based on these indications the processor displays the MOE for the fielding unit. The MOEs for the fielding units will be stored in one of the processor's data bases.

The processor asks for the purpose of the system. The processor reminds the user that the purpose of the system is in the MAA, O&O and concept paper. The purpose is asked for to focus and orient the user who is then given a menu of the 13 mission areas to choose from. The user indicates the most appropriate mission area for the processor which then recalls and displays the menu of functions and subfunctions appropriate to the selected mission area.

The processor reminds the user that the functions of the system are specified along with its purpose in the MAA, O&O and possibly concept papers. Then the processor prompts the user to choose from a menu of functions, those appropriate to the system. This is repeated for three levels of "functions." Given the appropriately limited menu of functions, the purpose of the system and its O&O and or concept paper, the choice of "functions" at each level is straight forward as will be seen below in an example.

After the processor presents all of the menus a final set of potential specifications are presented in conjunction with the lowest level of functions. The user makes the choices of specifications which are possible based on the purpose and concept of the system. Then the processor compiles from all of the menus the objectives of the system. The whole process is shown in the following example.

To provide consistency throughout the description of the processor, the Forward Area Ammunition Support Vehicle (FAASV) will be used as the example system. The FAASV is a full-tracked, aluminum armored, ammunition resupply vehicle with mechanized onboard ammunition handling equipment. It looks much like the M109A2 self propelled howitzer without the cannon. In fact, the FAASV is built on an M109A2 chassis. The FAASV is capable of carrying approximately 400 complete-rounds (projectile, fuze, propellant).

In the following sections describing the execution of the functions of the processor, quotation marks (" ") and all capital letters are used to indicate the user's input and processor's screen display. The text or numerical values shown within the quotation marks are intended to be illustrative of the actual user and processor responses; they are used only as a means of describing the use of the processor. Notes to the reader and descriptions of responses are contained in parentheses (). The sequence of events is depicted vertically with subsequent events occuring lower on the page or on subsequent pages.

****************** USER INPUTS AND PROCESSOR RESPONSES ******************

USER INPUT	PROCESSOR RESPONSES
(User accesses the processor)	
	"TYPE IN THE NAME OF THE SYSTEM AND THEN FROM THE LIST OF UNITS SHOWN BELOW, INDICATE THE NUMBER OF THE UNIT WHICH UNIT WILL FIELD THE SYSTEM." (A list of numbered units is displayed for which unit MOEs have been developed and stored in the

USER INPUT	PROCESSOR RESPONSES
	processor during its development. Also provided is a category labelled "other" which if chosen will lead cause the processor to tell the user that a unit MOE will have to be be developed. If "other" is chosen the processor asks the user if he wishes to have it develop a new unit MOE.)
"10" (An example response)	
	(The processor displays the unit MOE including its factors, weights, minimal scores, related conditions and assumptions, definitions of terms and application rules for the MOE. "TYPE IN THE DEFICIENCY THE SYSTEM IS TO FULFILL AND THE PURPOSE OF THE SYSTEM. THE PURPOSE IS STATED IN ONE OR MORE OF THE FOLLOWING: THE RELEVANT MAA; THE O&O PLAN; A CONCEPT PAPER."
(User types in both)	
	"SELECT THE MISSION AREA OF THE SYSTEM FROM THE FOLLOWING LIST." (The 13 mission areas are displayed.)
"3" (3 = Fire Support)	
	(Selection of the correct mission area calls forth the correct data base of functions and subfunctions to form objectives.)

USER INPUT

PROCESSOR RESPONSES

"FROM THE FUNCTIONS AND SUBFUNCTIONS SHOWN BELOW, SELECT THOSE THE SYSTEM WILL HAVE TO PERFORM. FUNCTIONS ARE ROLES THE SYSTEM IS TO PERFORM. FUNCTIONS OF THE SYSTEM MAY BE EXPRESSED IN THE MAA OR O&O PLAN AS MISSIONS OR OBJECTIVES OF THE SYSTEM.

"PRESS F2 FOR MORE EXPLANATION OF FUNCTIONS AND HOW TO IDENTIFY THEM."

Function:
Target Servicing

Subfunction:
____ Maneuver
____ Battle Control
____ Battlefield Preparation
____ Target Acquisition
____ Target Attack
____ Target Attack Assessment
____ Support/Sustainment
____ Training

Function:
Counterfire

Subfunction:
____ Maneuver
____ Battle Control
____ Battlefield Preparation
____ Target Acquisition
____ Target Attack
____ Training

Function
Interdiction

Subfunction:
____ Maneuver
____ Battle Control
____ Battlefield Preparation
____ Target Acquisition
____ Target Attack
____ Target Attack Assessment
____ Support/Sustainment
____ Training

USER INPUT

PROCESSOR RESPONSES

(User selects "target servicing" and under that "support/ sustainment" as the most appropriate. These choices are easy as the user knows the intended purpose and use of the system and given those the choices he made are obvious and the only logical ones.)

(The list of functions and subfunctions being used in this example are just suggestions. They are currently being used by TRADOC personnel in mission area analyses as "battlefield tasks" and "subtasks" and are shown in Appendix B. Several other sets of functions are currently being developed for the Army and are available for use instead of those listed.)

"FROM THE ACTIVITIES SHOWN BELOW, SELECT THOSE THE SYSTEM WILL HAVE TO PERFORM."

Activities:
1) Acquire Targets
2) Aviation Support
3) Combat Information and Intelligence
4) Command and Control
5) Communicate
6) Engineer Support
7) Maintenance Support
8) Medical Service
9) Move
10) Personnel Service
11) Shoot
12) Supply
13) Sustain

(This list of "activities" has been developed for further specifying the objectives of systems. Other lists could be used in lieu of this one.)

"5, 9, 12"

USER INPUT

PROCESSOR RESPONSES

(User selects "communicate", "move", and "supply." Again, these are the only logical choices given the purpose of the system and functions it is supposed perform.)

"TO FURTHER SPECIFY THE SYSTEM'S OBJECTIVES,SELECT THE APPROPRIATE CHOICES FROM THE FOLLOWING ACTIVITY SPECIFICATION MENU FOR THE ACTIVITY OF MOVE. PLACE Xs IN FRONT OF THE APPROPRIATE CHOICES. IF THE CHOICES CANNOT BE ACCURATELY AND RELIABLY MADE, DO NOT MAKE THEM."

(The following menu is one of a set for each of the activities. The entire set is shown in Appendix C. Others could be used or those in the appendix could be modified.)

Activity Specification

Why:
____ Maneuver
____ Tactical march
____ Convoy
____ Reposition/relocate
____ Deliver
____ Evacuate
____ Recon
____ Patrol
____ Liaison/visit
____ Other (__________)

What:
____ Personnel
____ System (w/crew)
____ Cargo, bulk
____ Cargo, water

Mode of Transport:
____ On foot
____ Ground
____ Air
____ Water

USER INPUT

PROCESSOR RESPONSES

Terrain:

____ Road

____ Cross country

Distance(km):

____ 0 - 10

____ 11 - 50

____ 51 - 100

____ 101 - 200

____ > 200 (specify: ____)

Fordability (in):

____ < 24

____ 25 - 26

____ 37 - 38

____ > 48 (specify: ____)

How Many/How Much:

_______ (# personnel estimate)

_______ (# systems)

_______ (# tons)

_______ (# gallons)

Frequency:

____ 1

____ 2

____ 3

____ 4

____ > 4 (specify: __________)

Protection:

____ None

____ Personnel

____ Cargo

Speed (km/hr):

____ 0 - 5

____ 6 - 40

____ 41 - 120

____ 81 - 120

____ > 120 (specify: __________)

(The user makes the appropriate selections based again on the purpose of the system, the MAA and the O&O. Given the menu and the limited choices, the user will be able to easily make the appropriate choices for almost all of the

USER INPUT

categories. However, some choices will not be possible during early stages so the user will avoid them. For example, in regard to the previous menu's category for speed, the user would not be able to reliably specify and so is cautioned not to do so.)

PROCESSOR RESPONSES

(The user is then asked to complete the same type of activity specification menu for the other two activities.)

"PUTTING TOGETHER ALL OF THE FUNCTIONS, SUBFUNCTIONS, ACTIVITIES AND SPECIFICATIONS, THE FOLLOWING OBJECTIVES HAVE BEEN FORMULATED FOR THE SYSTEM. CHANGE THEM IF YOU WISH.

Mission Area = **FIRE SUPPORT**
Function = **TARGET SERVICING**
Subfunction = **SUPPORT / SUSTAINMENT**

Activity = **MOVE**
CARGO
ON GROUND
X COUNTRY
200 KM"
(etc. for the other specs.)

(The other activities and their specifications also would be presented along with the higher level functions and subfunctions.)

Development of Performance Criteria for Systems

This section describes the development of performance criteria for the system. The basic concept relies on the experts to give judgements about the effect potential capabilities of the system might have on its unit's effectiveness. The process is shown in Figure 4. The user starts the process by indicating he wishes to have the processor develop criteria.

The processor takes over and recalls from one of its data bases a candidate list of characteristics that could serve as the basis for the

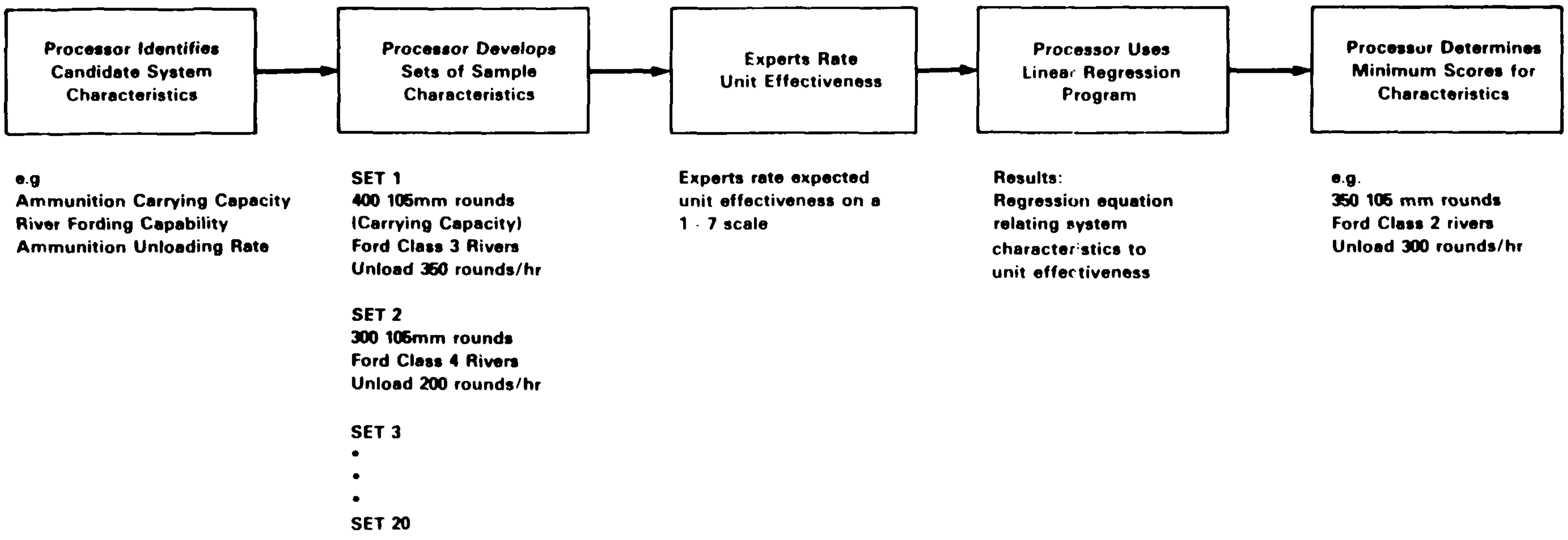

Processor Identifies Candidate System Characteristics
Processor Develops Sets of Sample Characteristics
Experts Rate Unit Effectiveness
Processor Uses Linear Regression Program
Processor Determines Minimum Scores for Characteristics
e.g
Ammunition Carrying Capacity
River Fording Capability
Ammunition Unloading Rate
SET 1
400 105mm rounds
(Carrying Capacity)
Ford Class 3 Rivers
Unload 350 rounds/hr
SET 2
300 105mm rounds
Ford Class 4 Rivers
Unload 200 rounds/hr
SET 3
•
•
•
SET 20
Experts rate expected unit effectiveness on a 1 - 7 scale
Results:
Regression equation relating system characteristics to unit effectiveness
e.g.
350 105 mm rounds
Ford Class 2 rivers
Unload 300 rounds/hr

Figure 4. Steps to develop system performance criteria.

system's performance criteria. For example, characteristics the processor could display for the FAASV might be speed, river fordability, or carrying capacity. The processor then asks the user if he wishes to modify the list of candidate characteristics and the baseline values associated with them. The baseline values are the performance values the predecessor system or similar system is capable of. For example, baselines value for the FAASVs characteristics might be speed - 30 mph; river fordability - class 4 rivers; and carrying capacity - 300 rounds.

Next the processor prompts the user to estimate alternatives to each of the baseline values. The processor instructs the user to rely on his knowledge of available technology and estimate the realistically largest and smallest values that possibly should be considered as performance levels for the FAASV for each of the candidate characteristics. His estimates even if inaccurate will be acceptable. They serve only as "ballpark" figures.

Then experts individually log on the system at their locations. The experts are queried by the processor to make judgements of the expected effectiveness of the unit. For each judgement the processor requests, the expert is told to consider the new system to be part of the unit. In addition, for each judgement, the experts are given a different set of of values for the candidate system performance characteristics. For example, the first two sample sets of characteristics with sample or candidate values might be :

1

- travelling speed - 30mph
- carrying capacity - 400 105mm rcunds
- river fordability - class 4 rivers

2

- travelling speed - 40mph
- carrying capacity - 350 105mm rounds
- river fordability - class 5 rivers

In addition, the unit will be described to the expert in terms of its OP order, deficiency, etc. The experts give their judgements of unit effectiveness on a one to 100 scale of unit effectiveness.

The processor collates all the experts judgements and performs a multiple regression analysis. It then indicates which characteristics were significantly related to unit effectiveness, the importance of each factor, and the relationship between the characteristics and unit effectiveness. The processor also determines minimum scores for each characteristic. The whole process is depicted in the following example.

******************* USER INPUTS AND PROCESSOR RESPONSES ********************

USER INPUTS	PROCESSOR RESPONSES
(The user logs on and identifies the system.)	

USER INPUTS

PROCESSOR RESPONSES

"THIS BEGINS THE PROCESS OF DEVELOPING PERFORMANCE CRITERIA FOR THE SYSTEM.

THE PURPOSE AND OBJECTIVES OF THE SYSTEM ARE THE FOLLOWING:"

(These would be printed on the screen and also automatically on hard copy unless suppressed.)

"TYPE IN THE NAME OF THE PREDECESSOR SYSTEM IF THERE WAS ONE."

"The M 548 truck."

(If there is no predecessor system, the processor asks the user for the names of similar systems.)

"THE PREDECESSOR SYSTEM AND SIMILAR SYSTEMS OF THIS TYPE USED THE FOLLOWING CHARACTERISTICS FOR PERFORMANCE CRITERIA:"

(The characteristics or factors that performance criteria were based on for the predecessor OR similar systems will be recalled from one of the processor's data base and displayed.)

"CONSIDER THESE CHARACTERISTICS AND THE

DEFICIENCY
MAA
O&O
CONCEPT PAPERS
AND THE MOE FOR THE FIELDING UNIT."

(MOE is displayed.)

"DELETE OR ADD THE NAMES OF ANY ADDITIONAL CANDIDATE PERFORMANCE CHARACTERISTICS YOU WANT TO HAVE CONSIDERED OR HIT RETURN IF YOU DO NOT WISH TO ADD ANY."

USER INPUTS

PROCESSOR RESPONSES

(User adds two characteristics.)

"THE FOLLOWING CHARACTERISTICS WILL BE PRESENTED TO EXPERTS AS THE BASIS OF A CANDIDATE LIST OF SYSTEM PERFORMANCE CRITERIA:"

(Set of characteristics presented to user plus his two additional ones are displayed.)

"NEXT TO MOST OF THE CHARACTERISTICS ARE BASELINE VALUES WHICH ARE THE MINIMALLY ACCEPTABLE PERFORMANCE SCORES FOR THE PREDECESSOR OR SIMILAR SYSTEM IDENTIFIED ABOVE. PLEASE MODIFY THOSE VALUES IF YOU THINK THEY SHOULD BE CHANGED."

(User makes response.)

(Headings for two new values appear on the screen next to the column for baseline values.)

"PLEASE ENTER TWO NEW VALUES FOR EACH CHARACTERISTIC:

- THE LARGEST WHICH IS THE REALISTICALLY GREATEST THE NEW SYSTEM COULD POSSIBLY ACHIEVE GIVEN THE STATE OF AVAILABLE TECHNOLOGY;

- THE SMALLEST WHICH IS THE REALISTICALLY LOWEST THAT YOU THINK THE UNIT COULD TOLERATE FROM THE SYSTEM. MAKE YOUR BEST ESTIMATES BASED ON YOUR EXPERT OPINION."

(User enters values.)

"IN ORDER FOR THE EXPERTS TO ESTIMATE THE EFFECT OF VARIOUS

USER INPUT	PROCESSOR RESPONSES
	POTENTIAL PERFORMANCE CAPABILITIES OF THE SYSTEM ON THE EFFECTIVENESS OF THE UNIT, THE EXPERTS WILL NEED TO KNOW THE CIRCUMSTANCES SURROUNDING THE UNIT. THE CIRCUMSTANCES INCLUDE THE MISSION, LOCATION, ENEMY ETC. SO MAKE UP AN EXAMPLE OF THE MOST TYPICAL MISSION THE FIELDING UNIT MIGHT BE GIVEN. IN OTHER WORDS MAKE UP A BRIEF OP ORDER. THE OP ORDER MUST CONTAIN ....."
(User types in OP Order which can be obtained from an ARTEP, O&O, or an expert.)	
	"NOW DESCRIBE THE ENEMY FORCES THE UNIT IS MOST LIKELY TO ENCOUNTER AND A LIKELY LOCATION OF THE OPERATION."
(The user types in the responses.)	
	"NOTIFY THREE EXPERTS THAT IT IS TIME FOR THEIR INPUT."
(Later, on their terminals, the experts individually logon the system and each goes through the following.)	
	"WELCOME. YOU WILL SUPPLY THE EXPERT JUDGEMENTS FROM WHICH TO DEVELOP PERFORMANCE CRITERIA FOR THE SYSTEM. IN ORDER TO GIVE YOUR JUDGEMENTS YOU NEED TO KNOW THE FOLLOWING:" (Mission is presented.) "YOU ALSO NEED TO KNOW THE FOLLOWING:

USER INPUT

PROCESSOR RESPONSES

(Presented here will be:
1. The unit MOE and how to use it.
2. The deficiency, purpose and objectives of the system.
After being shown all this information on the screen, the processor has a copy printed for them.)

"YOUR EXPERT JUDGEMENTS ARE NEEDED TO DEVELOP PERFORMANCE CRITERIA FOR THE FAASV. YOUR JUDGEMENTS ARE ALSO NEEDED TO IDENTIFY CONDITIONS THAT COULD AFFECT THE PERFORMANCE OF THE FAASV.

YOU WILL BE ASKED TO ESTIMATE THE EXPECTED EFFECTIVENESS OF A FAASV's BATTERY SEVERAL TIMES. EACH TIME THE POTENTIAL PERFORMANCE CHARACTERISTICS OF THE FAASV WILL BE SLIGHTLY CHANGED. WHEN MAKING YOUR JUDGEMENTS, CONSIDER THAT THE BATTERY WILL THEORETICALLY BE PERFORMING THE FOLLOWING MISSION AGAINST ______ IN THE AREA OF ____________.

IN GIVING YOUR JUDGEMENTS OF THE EXPECTED EFFECTIVENESS OF THE BATTERY, USE THE SCALE OF ONE TO 100 SHOWN BELOW WHERE 50 IS CONSIDERED MINIMALLY ACCEPTABLE ....."

(Scale presented to user.)

"FOR YOUR FIRST ESTIMATE OF THE EXPECTED EFFECTIVENESS OF THE BATTERY, CONSIDER THE FAASV TO HAVE THE FOLLOWING CHARACTERISTICS AND VALUES:

1) MAX CRUISING SPEED = 30 MPH
2) MAX CARRYING CAPACITY = 300 RDS
3) . . . . . .
n) . . . . . .

IF THE REST OF THE BATTERY PERFORMED AS PRESCRIBED, WHAT WOULD BE THE EFFECTIVENESS OF THE BATTERY

USER INPUT	PROCESSOR RESPONSES
	IN ACCOMPLISHING ITS MISSION? TYPE IN A NUMBER BETWEEN ONE AND 100.
"55"	
	(The same question is asked 20 times as the values of the system characteristics are changed each time.) "THAT WAS THE LAST JUDGEMENT YOU NEED TO MAKE. YOUR JUDGEMENTS WILL BE COMBINED WITH THOSE OF THE OTHER EXPERTS AND FEEDBEACK GIVEN TO YOU LATER. THANK YOU FOR YOUR ASSISTANCE." (After all the experts give their judgements the processor performs a multiple regression analysis. The processor regresses the various sample values used for the system characteristics - independent variables - on to the judgements of the experts - the dependent variable - i.e., the values of one to 100 for unit effectiveness. Then the processor performs significance tests for each factor.)
(User logs on and asks for results.)	
	"THE FOLLOWING CHARACTERISTICS WERE FOUND TO BE SIGNIFICANTLY RELATED TO UNIT EFFECTIVENESS:" (Characteristics are presented.) "THE REGRESSION MODEL RELATING THE CHARACTERISTICS TO UNIT EFFECTIVENESS IS THE FOLLOWING:" (Regression formula.)

USER INPUT	PROCESSOR RESPONSES
	"THE REGRESSION INDICATES THAT THE WEIGHTS AND THUS THE IMPORTANCES OF EACH OF THE CHARACTERISTICS ARE THE FOLLOWING:"
	(Importances of each factor shown.)
	"THE FOLLOWING IS A TABLE SHOWING THE RELATIONSHIP BETWEEN VARIOUS VALUES OF EACH SYSTEM FACTOR AND THE EFFECTIVENESS OF THE UNIT."
	(Table displayed.)

Identification of Conditions

This section describes the basic concept for developing operational, tactical and environmental conditions that may significantly affect system performance. The basic concept, shown in Figure 3, is so deceptively simple that its reliability and efficiency are not obvious. The basic concept relies on the same experts during their giving of estimates of unit effectiveness.

The process actually begins with the user just after he finishes giving "ballpark" estimates of alternatives to the system characteristic baseline values. The process of identifying conditions is being described now, separately from the earlier part of giving "ballpark" alternatives and developing system performance criteria, solely for sake of clarity. The process of identifying conditions actually occurs during the development of system performance criteria.

After the user gives his "ballpark" alternatives, the processor takes over and recalls from one of its data bases, a list of candidate environmental, operational and tactical conditions that are relevant to the fielding unit. The processor asks the user if he wishes to add to or modify the list of candidate conditions which the user then acts on. The user is then told to notify the experts. The user does so and logs off.

Then, as described before in the development of performance criteria, the experts log on the system to give their judgements of the expected effectiveness of the unit. The interaction between the processor and the experts begins exactly as it was described in the section on the development of system performance criteria. However, in addition to being asked for their judgements of unit effectiveness as described before, the experts are also asked to judge the effectiveness of the unit with the unit operating under each, taken one at a time, of the conditions of the list of candidate conditions.

Candidate system performance characteristic values. Instead conditions are used with only three sets of the system performance characteristic values.

After the experts give all of their estimates, the processor compares the experts' estimates of unit effectiveness given without the unit operating under a condition (e.g., nighttime) to the estimates of effectiveness given with the unit operating under the condition. If the results of the comparisons indicate that effectiveness is impaired when a condition is included in the contextual information about the unit, then that condition is deemed one that will probably affect the level of performance that the system can achieve.

Development of New Unit MOE

The processor will contain MOE for all types of units. Thus it is very unlikely that the user will ever have to develop a MOE for a unit. However, on the off chance that this might be necessary, the processor will have the capability to develop unit MOE.

The procedures for developing new unit MOE will be, as one might suspect, almost identical to those for developing criteria for a new system as the criteria for a new system amount to a system MOE. The data base of candidate characteristics for the new unit MOE will be the characteristics of the existing MOEs. The processor will present to the user the names of the types of units for which MOE are stored in the processor. The user will indicate which of these types of units are similar to the unit for which the processor is to develop a MOE.

From the similar units the processor will compile a list of characteristics used in the similar units' MOEs. These will serve as the candidate list of unit MOE characteristics for the MOE to be developed.

The remainder of the procedures are identical to those for developing system criteria. The user is queried by the processor to make any changes he deems necessary to the candidate characteristics. Then the experts are queried by the processor for their judgements of unit effectiveness. However, this time the processor does not vary values for system criteria as it does when it is querying experts in order to develop system criteria. This time the processor varies the values of each of the potential unit MOE characteristics (e.g., ground gained, casualties, enemy losses, etc.). The new system is not even mentioned at this time. Only the development of a MOE for the unit is the focus.

After the experts give their judgements the processor uses multiple regression to determine the relationship between the candidate unit MOE characteristics (e.g., ground gained) and the experts' judgements of unit effectiveness. The results are the unit characteristics (e.g., ground gained) which are significantly related to unit effectiveness, the weights for each significant factor, and the minimally acceptable scores for each factor.

MECHANICS OF THE PROCESSOR

This section describes in detail how the processor will perform each of its functions. The internal processes described include all the algorithms that are employed and the sequencing of all the processes. The descriptions of the workings of the processor are grouped under the headings of its four major functions: the development of objectives; the development of performance criteria; the identification of conditions; and the development of new unit MOE. The format within each of these sections is to briefly list the steps the processor will perform and then explain how the steps will be performed.

The Development of Objectives

The following steps summarize the processor's development of objectives (which are the same as the SOW's "missions").

1) Recalling and displaying the appropriate set of functions and subfunctions.

2) Instructing the user to make choices of functions and subfunctions and recording the choices.

3) Formulating objectives for the system from the choices of functions and subfunctions.

All of these functions can be easily programmed. The beauty of the process is embodied in the fact that the processor will contain a data base of the ingredients for developing objectives. The process numbered two is straight forward and requires no algorithms. The means of accomplishing numbers one and three, although very simple, are described in the following paragraphs. Each step is presented first (underlined) and followed by the explanation of how it will be performed by the processor.

1) <u>Recalling and displaying the appropriate set of functions and subfunctions</u>. The functions will be grouped into subsets. The organizational scheme will depend on which functions and subfunctions are selected as those the processor will use. A likely candidate set are those currently employed by users to do mission area analyses. They were developed by the schools and while they are called battlefield tasks and subtasks they are functions and subfunctions.

The TRADOC schools developed a group of functions for each mission area. For each function they developed a set of subfunctions. If these were adopted for the processor, the user's identification of the mission area of the new system would trigger the processor to rely on the set of functions developed for that mission area.

After the identification of the mission area of the new system, the processor would present a menu of only the functions of that mission area. After the user selects the appropriate functions based on the purpose of the system and its deficiency, the processor would then present in menu format the subfunctions developed for the previously presented set of functions. A third and maybe fourth level of subfunctions might be chosen in the same way.

3) <u>Formulating objectives for the system from the choices of functions and subfunctions</u>. The process of formulating objectives is a very simple one because the objectives will be compilations of the functions chosen. No natural language processing of the functions will take place so that the functions are merged into syntactically correct sentences. However, the processor will collate and present the functions and subfunctions selected. They will be presented in a decending hierarchical fashion with lower level functions presented below higher level ones. To accomplish this the processor will record the order in which the functions were chosen and display them in that order, indenting each lower level function as shown below:

Mission Area	=	**FIRE SUPPORT**
Function	=	**TARGET SERVICING**
Subfunction	=	**SUPPORT/SUSTAINMENT**
Activity	=	**MOVE**
		CARGO
		ON GROUND
		CROSS COUNTRY
		200 KM
		(etc. for the other specs.)

The Development of Performance Criteria

The processor will develop performance criteria for systems by performing the following steps.

1) The processor displays candidate system characteristics and their baseline values.

2) The processor records the user's additions to the candidate system characteristics, baseline values and estimates of the smallest and largest realistic alternatives to the baseline values.

3) The processor records the user's input of a high probability Op order, description of the enemy, and location of the mission.

4) The processor develops 20 sets of sample values for each of the candidate system characteristics.

5) The processor uses the 20 sample values for each characteristic to make and store internally 20 sets of candidate system characteristics.

6) The processor presents to the experts, the sets of sample characteristics of the system, one set at a time. The processor also presents the Op order, description of the enemy, location of the mission, unit MOE, deficiency, purpose of system and the

objectives of the system. With the presentation of each set of sample characteristics, the processor asks each expert, treated individually, for an estimate of the expected effectiveness of the unit. The experts are told to make their estimates on a scale of one to 100 with 50 (the midpoint) defined as minimally acceptable unit performance. The experts make their estimates by typing in a number between one and 100.

7) The processor performs a multiple regression analysis in which the values of the sample system characteristics are regressed on to the experts' estimates of unit effectiveness (which were given on the one to 100 scale).

8) The processor performs significance tests of the amount of variance accounted for by each of the candidate system characteristics. Those characteristics accounting for a statistically significant amount of the variance in unit effectiveness estimates are deemed significant.

9) The processor removes from consideration those characteristics deemed non-significant. Then the processor recalculates the multiple regression equation without the non-significant characteristics. The result is a regression equation with weights for each of the significant factors.

10) The processor searches the data for the minimally acceptable score for each of the significant factors. These are found and displayed with the regression equation, list of significant factors and the weights of each of the factors.

How the processor will perform each of these steps is described in the following paragraphs. Each step is presented first (underlined) and followed by the explanation of how it will be performed by the processor.

1) The processor displays candidate system characteristics and their baseline values. The user will be asked for the name of the predecessor system. The processor will contain the performance criteria for most systems that will probably be having successors built in the near future. The criteria will be grouped under the heading of the name of their system so that when a system is named by a user as a predecessor or similar system, the processor will immediately use the system's name to recall and display all that system's performance criteria.

The data base for predecessor systems will have the capacity of accepting new systems and their criteria. Also there will be a facility for having users easily enter the performance criteria of systems into the data base.

2) The processor records the user's additions to the candidate system characteristics, baseline values and estimates of the smallest and largest realistic alternatives to the baseline values. The processor does this by telling the user to type in these values. The processor tells the user to consider the deficiency as the prime supplier of candidate system

characteristics. The orocessor also tells the user to consider available technology and the unit's O&O as constraints. If the user is even moderately inaccurate, it will make little difference as these values are used only as starting points or "ballpark" figures from which to generate examples. The processor records user's smallest and largest estimates and ties them to the appropriate characteristics.

3) The processor records the user's input of a very high probability Op order, description of the enemy, and location of the mission. The processor displays the purpose of the system and its functions. Then the processor shows the user examples of each input. The correct format for these inputs is displayed with directions on how to make the imput. Then the processor records the user's input.

4) The processor develops 20 sets of sample values for each of the candidate system characteristics. Each set of candidate system characteristic values is used as the stimulus for the judgements the experts make of the expected effectiveness of the unit. The processor generates the 20 values for each system characteristic from within a range of values. The processor establishes the range as being equal to the values supplied as the largest and smallest alternatives to the baseline values. Working within this range, the processor divides the range up into 20 equal intervals by subtracting the smallest score from the largest score and then dividing the difference by 20. The resulting dividend is added to the smallest value plus the dividend 20 times to form 20 equal interval values between the smallest and the largest. All values are rounded to the nearest whole number.

As an example of this process for a single characteristic (note this is done independently for each characteristic), consider the following. For the characteristic of speed which has a baseline value of 30 mph, the values of 70 mph and 10 mph were established as the largest and smallest realistically possible alternative bounds. The processor subtracts 10 from 70 to get 60 and divides this by 20 to get 3. The dividend of 3 is added to the smallest value of 10 to yield 13, and then 3 is added to 13 to yield 16, and so forth 18 more times to yield 20 values beginning with 13 and continuing 16, 19, 22, etc.

5) The processor uses the 20 sample values for each characteristic to make and store internally 20 sets of candidate system characteristics. The 20 values for each characteristic are combined to form 20 sets of the same characteristics, each set having different values. The procedure for combining tha 20 values for each characteristic into 20 sets is to randomly select one of the 20 values for each characteristic and form a set from them. From the remaining 19 values for each characteristic, a random selection is made for each characteristic and these values form another set. The random selection is continued until all 20 values have been selected for each characteristic.

For example, the first two sets of values and characteristics for the FAASV might list the following as candidate values for the characteristics of speed and carrying capacity:

SET 1

Speed - 22 mph
Carrying Capacity - 200 rounds

SET 2

Speed - 46 mph
Carrying Capacity - 420 rounds

The processor makes only 20 sets of sample values rather than 50, for example, because the experts would probably not give more than 20 judgements. Time requirements and fatigue would preclude them from contributing more. Also, using three experts will yield 60 estimates which is enough.

The 20 different values used for each system factor are necessary to have enough values and thus degrees of freedom (df) to powerfully test the significance of the regression coefficients to be obtained. The statistical significance of the regression coefficients needs to be tested because they will be based on a "sample" of values and thus they might be capitalizing on chance. This needs to be ruled out. The coefficients need to be tested to determine if they are statistically different from zero.

6) The processor presents to the experts, the sets of sample characteristics of the system, one set at a time. With the presentation of each set of sample characteristics, the processor asks each expert, treated individually, for an estimate of the expected effectiveness of the unit. The experts are told to make their estimates on a scale of one to 100 with 50 (the midpoint) defined as minimally acceptable unit performance. The experts make their estimates by typing in a number between one and 100. The processor presents the sets of characteristics in the order in which they were developed. The processor records each response of the expert and pairs it with the values of the characteristics displayed at the time the response was given.

The 100 point scale is being used to give the experts enough range within which to make their estimates. Lesser ranges have constrained the experts.

The reason the sample sets of characteristics are being presented to the experts for their estimates is to obtain their rules for determining system criteria. In other words, the aspects of the system that are important to the unit. That the rules can be determined by presenting the experts examples and having them estimate unit effectiveness is a well established fact (Thomas and Cocklin, 1983; Dawes and Corrigan, 1974; Slovic and Lichtenstein, 1971; Darlington, 1968). That the experts cannot accurately and reliably tell the user their rules will be demonstrated in the following paragraphs. First the case for the reliability and validity of regression based modelling of expert's rules.

The modelling or capturing of experts' rules through the presentation of examples and the use of regression has a long history of providing easily obtained, extremely reliable (e.g., test-retest type regression coefficients of .95) algorithms embodying the decision rules of experts (Connelly, 1981; 1977; Darlington, 1968; Dawes, and Corrigan, 1974). Linear regression also has been shown to provide very valid representations of the cognitive models of experts (Slovic and Lichtenstein, 1971). Moreover, the validity of the approach has been demonstrated across many different content areas including the clinical judgments of psychologists (Goldberg, 1970), of radiologists (Hoffman et al., 1968) and the judgements of military experts (Thomas and Cocklin, 1983).

However, one might wonder, Why not just ask the experts to identify the criteria by simply and explicitly stating which of the system factors are significantly related to unit effectiveness? The reason this would not work is that the experts would not be able to reliably identify the significant factors and could not even come close to reliably stating the weights that should be accorded to the factors. The reason for the experts' inability to do this is that the experts are not totally aware of the components nor the dynamics of their cognitive models or decision rules. They can give estimates of the potential effectiveness of a unit which incorporates a new system (given all the other systems of the unit are described as performing satisfactorily). But the experts will be unable to reliably and comprehensively describe the rules they used to make their estimates of unit effectiveness.

This is not surprising if one considers how even the simplest of cognitive models operates. For example, most people would consider themselves expert on being able to identify an animal as a cat; that is, to recognize a cat. However, if asked to state the rules they used to identify an animal as a cat, most people would be able to give only the most sketchy of descriptions which would not embody all of the components they used nor the ways in which the components were used and weighted.

For example, an explanation of the rules to identify an animal as a cat might begin with the statement that cats have a head, four legs, fur and ears. However, all of these characteristics also apply to dogs and lions and yet people can almost always recognize a cat when they see one. Further explanations of the cat recognition rules might include that cats have tails and are lightweight, but the application of these criteria would still not exclude raccoons and small dogs. Obviously even the simple cognitive model or rules for identifying cats contain many components and are very complex in the way the components are weighted. All of these are reasons for using regression and examples of system characteristics for the purpose of identifying the cognitive rules of the experts.

7) The processor performs a multiple regression analysis in which the values of the sample system characteristics are regressed on to the experts' estimates of unit effectiveness (which were given on the one to 100 scale). The regression analysis uses a linear least squares approach. The linear approach has been shown to account for most of the variance of the dependent variable in the modelling of experts' rules (Dawes and Corrigan, 1974). In

fact, several researchers have shown that a linear approach accounts for upwards of 90% of the variance of the dependent scores (Thomas and Cocklin, 1983; Connelly, 1981).

The regression analysis is performed is by finding a linear function that relates the values of the candidate system characteristics to the estimates of unit effectiveness. The regression algorithm of the processor uses a running summation of dependent, independent, squared and cross product terms instead of a matrix formulation to calculate the least squares fit of a mathematical function to expert's estimates of unit effectiveness. The running summation simply adds to each sum as data is read from a file and does not store all data in memory-- only the necessary sums are maintained in the computer memory. The advantage of the approach is that regressions for any size data file can be computed with little computer memory required.

The regression analysis results in an equation relating the candidate system factors to the experts' estimates of unit effectiveness. The equation is of the standard form

$$(a_1 + b_1x_1) + (a_2 + b_2x_2) + (a_n + b_nx_n) = \text{unit effectiveness}$$

where a equals a constant, b equals a coefficient and the xs are the candidate system characteristics. The coefficients are the same thing as "weights" for the characteristics. They indicate the relative importances of the characteristics. The equation is kept internal to the processor for its next step.

8) The processor performs significance tests of the amount of variance accounted for by each of the candidate system characteristics. Those characteristics accounting for a statistically significant amount of the variance in unit effectiveness estimates are deemed significant. The processor tests the significance of each of the candidate system characteristics by testing the significance of the absolute increment in the proportion of variance accounted for by each candidate system characteristic. This is done while holding constant the contribution to the variance by all the rest of the candidate system characteristics. The F test used to do this is of course also a test of significance for the part correlation between the candidate system characteristic and the unit effectiveness scores. The F test for each candidate system characteristic is

$$F = \frac{r^2{}_y\ (j \cdot g)}{(1 - R^2{}_{y.H})/(N - K - 1)}$$

which is the proportion of variance accounted for by a candidate system characteristic divided by one minus the total variance accounted for by all the other variables times the degrees of freedom.

9) The processor removes from consideration those characteristics deemed non-significant. Then the processor recalculates the multiple regression equation without the non-significant characteristics. The result is a regression equation with weights for each of the significant factors. The processor uses the same processes procedures it did before to perform a multiple regression. However this time it does not include the variables deemed non-significant. The result is another regression equation of the same form shown previously.

10) The processor searches the data for the minimally acceptable score for each of the significant factors. These are found and displayed with the regression equation, list of significant factors and the weights of each of the factors. To find the minimally acceptable score for each of the significant factors, the processor first identifies all those judgements of unit effectiveness given by the experts in which the unit effectiveness scores were at least 50 (which was defined as "minimially acceptable unit performance"). These unit effectiveness scores and the related sets of candidate system characteristics that the experts relied on when making the estimates are made into a subset of the data. Then working with only the subset of the data, for each of the candidate system characteristics, the processor identifies the smallest value in the subset. For each candidate system characteristic, the smallest value is adopted as the minimally acceptable score. This makes sense because these scores were the lowest the experts considered possible for the candidate system characteristics when they estimated that expected unit effectiveness would be at least a 50 (minimally acceptable).

Then the processor displays and prints out the following:

1) the regression equation and an explanation of it;

2) the list of significant candidate system performance characteristics which are now called "system performance criteria" and an explanation of what significance means;

3) the importances of each of the system performance criteria (their weights which are taken from the equation);

4) the minimally acceptable score for each of the system performance criteria;

5) a statement that the user can see a display of all the unit effectiveness scores above 50 along with their related sets of system charactersitic values.

The Identification of Operational, Environmental and Tactical Conditions

The processor will identify conditions by performing the following steps which are explained in subsequent paragraphs.

1) The processor recalls and displays the operational, tactical and environmental conditions that are relevant to the system's unit.

2) The processor asks the user if he wishes to change the list of conditions and the processor records the changes.

3) The processor queries the experts for estimates of the expected effectiveness of the unit while operating under each of the conditions.

4) The processor then compares the estimates of unit effectiveness given with conditions to those given without conditions. Those conditions that result in lessened unit effectiveness scores are deemed significant by the processor and displayed to the user with their related unit effectiveness scores.

1) The processor recalls and displays the operational, tactical and environmental conditions that are relevant to the system's unit. Each unit for which an MOE will be developed will also have stored in the processor a set of operational, tactical and environmental conditions. These will be obtained during the development of the processor from the unit's O&O, TOE, and ARTEPS, along with expert opinion.

2) The processor asks the user if he wishes to change the list of conditions and the processor records the changes. The processor provides a format with spaces for adding new conditions and an edit capacity for changing or deleting conditions already on the list. The processor tells the user where the conditions were obtained and the purpose of identifying them.

3) The processor queries the experts for estimates of the expected effectiveness of the unit while operating under each of the conditions. The procedures the processor follows for the interactions with the experts are identical to those followed when the processor asks the experts for estimates of expected unit effectiveness without conditions. However, the processor does not present the 20 sets of system characteristics for each condition. When describing the system with a condition, the processor uses only three of the sets of candidate system characteristics. The three sets are randomly selected. Thus each expert is asked for three estimates of unit effectiveness for each of the conditions being tested.

4) The processor then compares the estimates of unit effectiveness given with conditions to those given without conditions. Those conditions that result in lessened unit effectiveness scores are deemed significant by the processor and displayed to the user with their related unit effectiveness scores. The processor calculates two sets of mean unit effective scores to compare to identify the conditions which influence unit effectiveness. The first set of mean unit effectiveness scores are calculated from the estimates the experts gave when no conditions were discussed. The second set of mean unit effectiveness scores are based on those scores obtained when conditions were included in the description of the unit's situation. Both sets of means are based on the unit effectiveness scores the experts gave in conjunction with only the three sets of characteristics selected as described in step three.

After the processor calculates the arithmetic means, the processor compares the means with F tests. The conditions associated with the F tests that result in significant differences are then deemed to influence the effectiveness of the unit. These conditions are displayed to the user and deemed to significantly effect unit effectiveness.

The Development of New Unit MOE

The following are the steps the processor will perform to devevlop new MOE for units. However, recall that the processor will contain MOE for units. These will be synthesized during the development of the processor and stored in the processor for use by the experts. On the off chance that a new unit MOE will be needed the following will be performed by the processor.

1) The processor will present a list of the names of the units for which it has MOE. The processor will prompt the user to identify the names of the units which are similar to the one for which the processor will develop a new MOE.

2) The processor recalls and displays a set of candidate criteria that are the basis for the criteria of a new MOE.

3) The remaining procedures are the same as those for developing performance criteria for systems.

The following describes how the first two steps will be performed.

1) _The processor will present a list of the names of the units for which it has MOE. The processor will prompt the user to identify the names of the units which are similar to the one for which the processor will develop a new MOE_. The list of names will be generated during development of the processor and will consist of those units selected for the development of a MOE. The list will be expandable. Also, the name of each unit for which a new MOE is developed will be added to the list of names of units. This will be accomplished by a subroutine attached to the end of the new MOE development process. The subroutine will automatically add the name of the unit to the list of other units.

2) _The processor recalls and displays a set of candidate criteria that are the basis for the criteria of a new MOE_. The processor displays all the criteria that are part of the MOE for the units identified by the user. Recall that in the previous step the user identified units as similar to the unit for which the processor will develop a new MOE. All the criteria of each MOE in the processor will be resident in a data base. The data base will have files named for the units for which there are MOE. In these files will be the appropriate unit MOE criteria.

RELATIONSHIP OF THE PROCESSOR TO COMBAT MODELS

The output of the processor the user will be most in need of validating will be the system performance criteria. There are two fundamental approaches to validation of system criteria; fight the battle and measure effectiveness or compare the criteria to the results of some combat model. The first approach is the ultimate validation and in reality all other "criterion" measures are predictors of this outcome. However, to validate the criteria and thus provide the combat developer assurance that he has identified appropriate requirements, we shall explore the possibility of using combat models to validate the system criteria.

In discussing combat modeling as a validation approach we will identify several candidate models, discuss their pros and cons, and recommend one or more of them for use. During the course of the discussion the issues involved in translating product output into a form acceptable to the model will be discussed. Finally, a short discussion of logistics and real world considerations pertaining to use of the model will be provided.

In order to provide a recommended model one must first construct selection criteria. The appropriate criteria appear to be that the model must have a high enough level of resolution to relate to the system performance characteristics to be validated. The model also must be reasonably quick to run (four to six months) and not be too labor intensive. In addition, it must possess the confidence of the Army community so necessary to its function as a validation measure.

In general a combat modeling approach to criterion validation has much inherent appeal because it is a highly valued approach among members of the Army ORSA community and partly because it has the ability to provide quantifiable results. Most Army models provide results in the form of:

- force ratios (number of Red forces/number of Blue forces)
- loss exchange ratios (number of original Red forces - number of remaining Red forces/number of original Blue forces - number of remaining Blue forces)
- killer/victims scoreboards (matrix of red and blue assets with kill ratios in the cells)
- territory gained and lost

When these kind of results are plotted out on a terrain board and interpreted by a team of tactical experts, they provide the basis for determining the effectiveness of the forces.

Army models come in a variety of levels of resolution from single sensor propagation models all the way to theater level combat models. Our concern is with force on force combat models into which we propose to enter the values representing mission success criteria. The use of force on force models will allow for the determination of the success of notional systems against a "standard" threat under "standard" conditions. The resaarch question is basically "Can a system that achieves the criteria developed with SRDM win against the postulated threat?"

In order to identify Army Combat models, we reviewed the Catalog of Wargaming and Military Simulation Models (Guirreri, 1986). The 10th edition of this catalog lists descriptions of tools in general use throughout the DOD and the defense establishments of NATO countries. The catalog was developed with direct inputs from defense analysis agencies, contractors, research organizations and previous catalogs. There are some 600 models described from which seven were selected for further review. The basis for their selection was composed of several parameters, including force on force play, battalion level, high resolution, ease of use, systematic use, level of interactiveness, length of run time and general acceptance among the Army ORSA community:

- ARTBASS - Army Training Battle Simulation System
- CARMONETTE - Computer Simulation of Small Unit Combat
- CASTFOREM - Combined Arms and Support Task Force Evaluation Model
- CATTS - Combined Arms Tactical Training Simulation
- CORDIVEM - Corps and Division Evaluation Model
- FORCEM - Force Evaluation Model
- VIC - Victor in Commander

Subsequent to this review ARTBASS, CATTS, and CORDIVEM were dropped from further consideration. ARTBASS and CATTS are both training simulations and therefore are extremely labor intensive. They are computer aided training exercises not self contained simulations. Similarly CORDIVEM is no longer in use because of its drain on human resources. It requires a team of blue players and red players, a controller team and other attendant personnel. Usually 40 - 50 people are required for a model run. In addition, CORDIVEM is completely dependent on subjective interpretation of players which allows no control over the model's input parameters. The other four models are not without their problems but do warrant further description.

CARMONETTE is used to analyze battalion-level combat doctrine and tactics. It is a computerized, stochastic, event-sequenced simulation of a combined arms air or ground war game. It is played on a variable terrain representation of grid squares at 100 meters resolution for an hour of combat engagement. Force representation of infantrymen or various vehicles including tanks, armored personnel carriers, air defense, and helicopters, is at the individual up to platoon group size in a battalion-level force. Events pertaining to surveillance consider the effects of battlefield obstructions including weather, aerosol smoke, and artillery dust. Probabilities of hit and kill consider the biased dispersion of weapon systems based on moving shooters/targets. Output consists of displays and detailed reports including the killer/victim scoreboard.

As input CARMONETTE requires troop lists, weapon lists, weapon accuracy, weapon performance data, weapon lethality, sensor performance data, vehicle mobility characteristics, vehicle vulnerability, tactical scenario and terrain characteristics. CARMONETTE output lists all events assessed, with a summary of all casualty events and a summation of kills by target type and weapon types. Also available are summaries of weapon engagements (firings) shown by target type, rounds fired, personnel killed, and vehicles destroyed for each of the selected range brackets. Although CARMETTE has most of the qualities required above it was rejected because of time constraints and because it does not account for combat support.

VIC is a computerized, analytical, mid-intensity model developed as a replacement for CORDIVEM. It is used in estimating net assessments while performing force deployment studies, and in generating information for performing trade-offs among weapon systems. The outcome of force interaction is determined in terms of the ground gained or lost and attritions of personnel and weapons systems. The VIC model is a two-sided, deterministic simulation of integrated land and air combat. The level of aggregation is the maneuver battalion or its equivalent. It employs forces up to the level of a U.S. Corps facing an enemy of strength determined by scenario and theater in which the simulation takes place. VIC is an event-stepped model which also employs time steps for scheduling some actions. By this we mean that realistic events during the course of the battle drive the calculations of subroutines of the model. If nothing occurs for a predetermined period the time, step option automatically become operative. It uses modified differential equations for combat outcomes based upon the VECTOR-2 Model. Tactical decisions and force employments are determined by tactical decision tables supplied by the user to provide flexibility in controlling model processes. Each side may employ maneuver unit weapon systems and weapons of tactical aircraft, as well as artillery, mines, helicopters, air defense systems, and other means of conducting combat at the U.S. Corps level.

As input VIC requires forces and supply inventories, basic weapons performance data, other system performance data, geographic and terrain data, and tactical decision tables. Its outputs include casualties and systems losses (killer/victim scoreboards, etc.), FLOT traces and force positions over time, target acquisition and intelligence summaries, availability and condition of forces and supplies, and air battle and air defense results. VIC, although a well respected model, is designed for corps level operations and therefore has a resolution which is too low to accept system performance characteristics as input.

At the highest echelon level FORCEM provides simulation of all of the air-land activities in a theater of operations over an extended period (up to 180 days). Combat operations are at the division level and all of the combat support and combat service support functions from the port to the FLOT are represented. It is a fully computerized simulation used in studies and analyses of force planning and resource allocation issues. The model is also part of the AMIP. The model provides an average value, two-sided, time-stepped representation of the theater activities. Presently the minimum time cycle is a 12-hour period. The level of resolution for combat units is the division. Combat support and combat service support operations are represented by smaller organizational elements. Road, rail, and water transport routes are given a network representatior and terrain features are resolved to grid square; the size of the squares may be set as desired (5 to 30 km). Functional submodels represent the major activities of target acquisition, communication, command and control, division engagement, fire support, air operations, unit movement, and combat service support. As an average value simulation, without player interaction, command and control is represented by automated decision processes at three levels in the theater (corps, Army group, theater). Assessment of division battle is made through an analytic representation of a division engagement. The representation is done with sets of attrition coefficients calibrated to the results of engagements simulated by an independent division model.

FORCEM requires in-theater force units and assets; arrival schedule units and assets; theater scenario and plans; terrain; engagement results from division level simulation; weapons and equipment characteristics; C2 decision criteria; performance factors for surveillance, communications, repair, medical, transport, and engineering functions. Its output includes the status of units and assets over time, computer graphics and map displays, hard-copy plots and charts. Because of its theater level emphasis it is not appropriate for SRDM output validation. It is also quite time consuming to run.

CASTFOREM is intended to be the lowest echelon member of a hierarchy of models being developed as part of the Army Model Improvement Program (AMIP). The family of models will include battalion, division, corps, and theater-level force-on-force simulations. CASTFOREM meets all of the selection parameters set above and therefore is recommended as the most appropriate model with which to validate SRDM output.

Similar to CARMONETTE, CASTFOREM is a stochastic, event-sequenced, opposing forces simulation of ground combat involving up to a Blue battalion task force and a Red regiment. This model however, can be used in either batch or interactive modes with variable unit resolution down to the individual weapon system level. Resolution of terrain is also variable. Battlefield environments to be modeled include static weather, dynamic obscurants (smoke and dust), nuclear and chemical contaminants, and electronic warfare. In addition to fighting troops, all combat support and combat service support units and functions which interact with and affect the combat activities of maneuver units are represented in the model. Thus a very high level of resolution is possible. The model contains the command control logic in the form of decision tables to make tactical decisions which generate orders, reports, and request for support. These decision table outputs, in turn, control the actions of the units playing in the simulation.

CASTFOREM accepts as inputs many kinds of data, which also allows for a high level of resolution. These include terrain description parameters, weapon effects data, unit orders, CS and CSS equipment data, communications data and network structures, environment data, decision tables, organization structures, and personnel description parameters. As output each combat event is recorded for postprocessing.

Based on the parameters for model selection discussed earlier and on the ARI emphasis on the battalion task force level, CASTFOREM is the appropriate model with which to validate SRDM derived system success criteria. CASTFOREM accepts system performance characteristics which will influence the battle outcome. Thus testing the validity of the system performance criteria output from SRDM. CASTFOREM will take into consideration: 1) Line of sight/acquisition; 2) Resupply issues; 3) Ammunition effects; 4) Fire control issues, but may not accept attrition data which could come from another resiliency analysis model called AURA.

According to Keyes (1980) the Combined Arms and Support Task Force Evaluation Model (CASTFOREM) is a product of TRAC White Sands Missile Range, New Mexico. It is intended to be the lowest echelon member of a hierarchy of models which includes Theater, Corps, and Division level force-on-force simulations. It is being develop to satisfy two user requirements:

- Elimination of shortcomings in existing ground combat force-on force models and concomitant improvement of the quality and timeliness of studies.
- Support of analyses in TRADOC identified mission areas.

It is anticipated that the scenario preparation process for a CASTFOREM simulation will closely approximate the military planning process for a tactical operation in terms of both methodology used and man-hours required. This will be accomplished through the construction of sets of decision tables, for both RED and BLUE, each of which is designed for a specific type tactical operation (e.g. active defense, deliberate attack, hasty river crossing). The model contains doctrinal responses to a broad spectrum of tactical situations, requires user threshold inputs to trigger each doctrinal response and permits dynamic maneuver by opposing forces. Therefore the scenario development, data entry and run time should not exceed six months.

In order to give the notional units fighting the battle a procedure for decision making, they are provided access to a mental process module (MPM) and a physical process module (PPM). These modules provide potentially accessible ports for the input of human performance related mission success criteria. The mental process module will carry out the functions of analysis, planning, and decision making and will control the performance of the physical process module of the same name. For example, the engage mental process module command (Engage MPM) will control the decision making process pertaining to target selection, ammunition selection, firing time, and other related factors. The engage physical process module command (Engage PPM) will compute round impact time, hit probability, kill probability, etc. Similar mental and physical process module pairings exist for all other functions that a unit of resolution might perform (e.g., communication, search, engineer, resupply, etc.). The coordination of these functions is performed by the command and control modules to which each unit of resolution has access. Using these sophisticated programming techniques has resulted in a model which does not require inordinate investment in manpower.

Prior to commitment to this recommendation or the use of any other model, time and logistical requirements must be reconsidered. For example, although CASTFOREM appears to be the most appropriate model for linkage with SRDM, and was in fact developed to resolve the deficiencies of CARMONETTE (a monto carlo simulation), it may require months to load the data and run the simulation. Moreover, this time estimate only holds if the entry data fit the model constraints and if one of the canned scenarios is appropriate for the validation effort.

DEVELOPMENT OF THE PROCESSOR

Overview

The Product One lifecycle validation, verification, and testing (VV&T) process, including software development and maintenance will ensure the software quality, user acceptance, and ease of use. The VV&T process is a procedure of review, analysis, and testing used throughout the software lifecycle. Validation determines the correctness of the final program or software with respect to the software requirements. Verification employs integrity and evolution checking to determine internal consistency and completeness. Testing, either automated or manual, examines program behavior by executing the program on sample data sets. The software lifecycle is the period of time beginning with the software concept development and ending when the resultant software products are no longer available for use.

The software validation, verification and testing cycle is broken into five phases: requirements determination, design, programming and testing, installation, and operations and maintenance (Federal Information Processing Standards Publication, 1983). These five phases represent milestones in the software development process, and provide excellent points for user inspection. In addition, use of these five phases improves direct project management. Software developers and maintainers have a well defined set of tasks to perform. Verifiers, by checking the products of these tasks, can verify that the project requirements are met at each of the following five phases.

The five phases are outlined below.

Phase I. Requirements Definition and Analysis

- Developmemt of the project verification and validation plan
- Generation of requirements-based test cases
- Review and analysis of the requirements
- Review and analysis of the draft user manual

Phase II. Design

- Completion of Verification and Validation plan
- Generation of design-based test scenarios
- Design processor
- Review and analysis of the design
- Premliminary design integrity check
- Preliminary design evolution check
- Development of test support software

Phase III. Programming and Testing

- Completion of test case specification
- Write code
- Review, analysis, and testing of the program

- Code integrity check
- Code evolution check
- Unit test
- Integration test
- System test

Phase IV. Installation

- System acceptance

Phase V. Operations and Maintenance

- S)ftware evaluation
- Software modification evaluation
- Regression testing

Product 1 will undergo the first four phases of software development, but will not be maintained over the long term under the projected contract. The first four phases of development are therefore described below.

Phase I. Requirements Definition and Analysis

This phase consists of four parts: development of the project validation, verification, and testing plan; generation of requirements-based test cases (scenarios); review and analysis of the requirements, and review and analysis of the draft user manual(s).

The project plan explains the strategy for managing the development of the software. This document defines the general software development process for all phases of the project, estimates resource requirements, and specifies intermediate milestones, including management and technical reviews. It defines methods for design, coding, verification, validation and testing, document reporting, and change control. A basic set of test cases will be developed to clarify and to determine measurability of each software requirement. The acceptance criteria developed during evaluations by subject matter experts (SMEs) will be used to develop the test cases. Input data and expected results for each test case will be included in the specification.

The software requirements document will specify what the system must do, including the requisite information flows, processing functions, performance constraints, and the acceptance criteria for deciding that specific requirements have been satisfied. In addition, this document will also contain those internal specifications which, although transparent to the end user, are necessary for the development of the end product. This activity ensures that the requirements result in a practical, usable solution to the appropriate area.

Analysis techniques in the requirements phase include static and dynamic analysis. Static analysis focuses on checking adherence to specification conventions, consistency, completeness, and language syntax. Dynamic analysis focuses upon information flows, functional interrelationships, and performance requirements. Manual methods such as inspections, peer reviews, and "walkthroughs" are effective in accomplishing both types of analysis.

A user manual and tutorial script will be drafted. Howver, the manual will in no way lessen or compromise the embedded training the processor will contain. The manual will describe software system use in non-technical language. Each manual will describe both the system functionality and the user interface. Manual preparation during the requirements phase is an excellent mechanism for ensuring that both the users and the developers share the same view of the system. The manual serves as a reference document for the preparation of input data and as a useful tool in setting parameters for interpretation of the results. The users' manual will be reviewed for clarity and consistency. It will be checked for completeness against the requirements document. In addition, this verification activity will ensure that the internal specifications of the requirements document are defined sufficiently to produce the functions and interfaces described in the users' manual.

Phase II. Design

The goal of this phase is to design a solution. Alternative solutions are formulated and analyzed, and the best solution is selected and refined. A high-level specification which defines information aggregates, information flows, and logical processing steps is generated and refined into a detailed specification describing the physical solution (algorithms and data structures). The result is a solution specification that can be implemented in code with little additional refinement.

The design specification contains two documents: (1) a preliminary design document to identify a high-level solution developed during this phase and (2) a detailed design document that defines and refines software (algorithms and data) to be coded in the subsequent phase. The design will be analyzed to ensure internal consistency, completeness, correctness, and clarity, and to verify that the design, when implemented, will satisfy the requirements.

As initial design specifications reveal incorrect, inconsistent, infeasible, or ambiguous requirements, revised requirements specifications need to be developed. New or revised system requirements may warrant revision of the verification, validation, and test plan. The detailed design plan may indicate the need for additional testing procedures. Additional test scenarios and test cases (input data and expected results) will be developed to exercise and test logical and structural aspects of the design. Development or acquisition of any support software needed for unit, integration, or system testing will be completed and installed during the detailed design phase to ensure readiness during programming and testing.

Design specification schemes provide mechanisms for specifying algorithms and their inputs and outputs in terms of modules. Inconsistencies in specifying the flow of data through the modules can be detected by static analysis techniques. Dynamic analysis of design is accomplished by some form of design simulation. This will include a manual "walkthrough" and an automated simulation using a model of the design.

Phase III. Programming and Testing

During this phase, the detailed design is implemented in code, resulting in a program or system ready for installation. Three types of testing are performed: unit, integration, and system. The support programmer is responsible for unit testing, the responsibility for integration and system testing will be the responsibility of the core project team.

Unit testing checks for typographic, syntactic, and logical errors. Programmers will also check code modules to ensure that each correctly implements its design and satisfies the specific requirements. Static analysis techniques and tools are used to ensure the proper form of programming products, for example, code and documentation. Dynamic analysis techniques are employed to study the functional and computational correctness of the code. Initially, such manual techniques as "walkthroughs" can be used as an effective forerunner to testing.

Integration testing by the project manager focuses on checking intermodule communication links and on testing aggregate functions formed by groups of modules. Further, system testing examines the operation of the system as an entity, and in a simulated environment. This ensures that the software requirements have been satisfied both singly and in combination with other "real life" variables typically found in the end users environment. Sample data will be used in testing initial prototypes. Evaluation will include criteria selection, procedures, decision rules, and algorithms chosen.

The final activity of this phase is to ensure readiness of the software installation, including revision of plans as necessary and completion of all other coding, testing, and documentation. Fully documented and tested code is constructed and prepared for installation. Manuals describing the input and report formats, user commands, error messages, and instructions for operation by the user are completed. Final revisions and additions to the test data are made. Based on prototype testing, recommendations for revisions are made with extensive subject matter input (SME) input. Retesting is done to ensure confidence in the results and demonstrate ease of use. Actual results are compared with expected results and are validated for satisfaction of the requirements. These results are documented. Observed problems are recorded in formal statements and may necessitate returning to a previous phase for resolution.

Phase IV. Installation Phase

This phase primarily evaluates and modifies software, if necessary, to ensure user acceptance. To accomplish this, the system is placed in operation. The final task, integrating the system components, may include installing hardware, installing the program on the computer, reformatting/creating the data base and verifying that all components have been included. Modification of the program code may be necessary to obtain compatibility between hardware and software, or between different software modules for which earlier simulation testing may not have been adequate. The installation report will describe the results of the installation activities, including data conversion, and software/system problems and modifications. User acceptance and prototyped ease of use will be validated.

The next task is to test the system in its complete operating environment. The test data from earlier phases are enhanced and used. The result is a system qualified and accepted for production use. The installation report will also include the results of the testing conducted.

Next is the start of system operation. Interfacing with on-going software systems will be a prime consideration to save money and computer storage space. This task also includes operator and user training.

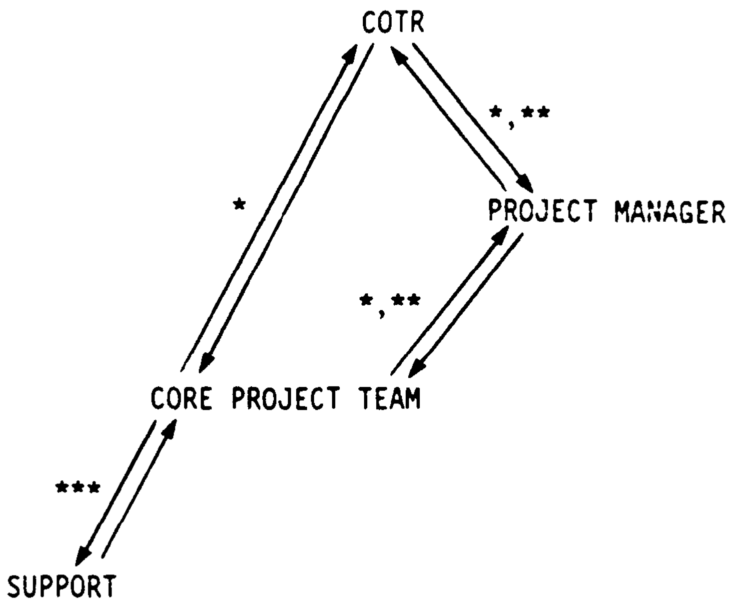
In summary, software development will consider the mission, the user, the data base and the algorithms so that user acceptance of output is maximized and difficulity of use is minimized.

Organization of the Project

Organization of the project for development of product 1 is described in Figure 5 and is based on the recommendations of Fred Brooks (1979) in his book "The Mythical Man-month". The relevant concept is that the fewest minds as possible should be assigned a software development job. As shown in the figure, formal and informal communication with core project team, the COTR, and project management permits direct interaction of all those individuals as well as the formal reporting tasks. The core project team consists of Dr. Joseph Conroy and Edward Connelly who are responsible for:

- requirement definition;
- user coordination during design;
- design;
- program code and unit test;
- documentation preparation and test;
- tutorial development;
- integration and system test;
- installation and user acceptance; and

As shown in the figure, work will be assigned to support personnel who will code processor modules and conduct module tests.



* Informal Working Coordination
** Formal Communication (Reports)
*** Task Assignments to Support Staff

Figure 5. Organization of the project.

How Product 1 will be Developed: Specific

Product 1 will include unit MOEs in addition to the processor, processor documentation, and tutorial. MOEs will be developed in parallel with the processor development, which is possible because the computational algorithm for producing MOEs presently exists and is routinely used to produce MOE's for various applications. Thus, the processor development work will consists of:

- simplification of the processor to provide just those functions needed,
- involvment of combat developers (or retired combat developers) in the initial requirements definition process,
- design of the human/machine interface,
- processor testing,
- documentation preparation and test,
- tutorial development,

- synthesis of unit MOE's and
- processor delivery and installation.

The previous section, titled Overview, gives the development plan in terms of processor testing. A description of the method of developing the product one processor and the unit MOE is given in the following paragraphs.

Requirements Definition

Combat developers (active or retired) will be interviewed to identify the computing equipment they typically use and their specific needs for MOEs. Individual CDs will be identified to evaluate alternative designs. Since the processor is essentially an instrument to collect data from experts in a setting away from the CD, the processor interface and instructions for the expert will be designed to be self sufficient.

A method for testing preliminary interface designs without requiring coding of each design is called the "wizard" system. With this system a display for a typical user (for instance retired army officers acting the part of CDs and SMEs) is driven not by the processor but rather from the keyboard entries of another person, known as the wizard. Interface designs are tested by having the wizard intitially callup a display which is presented on the user's screen. Then as the user responds, the wizard can answer in an intelligent way giving data, instructions, answering questions etc. As the interaction proceeds, which is a simulation of the interaction that could occur with the processor, all entires from both the "user" and wizard are recorded. When several users have tested with the initial design, the entries can be examined to detemine what responses the processor will have to be capable of providing and what instructions were not clear. Results of this "live" simulation generate, in a more or less automatic way, many of the processor requirements and also a provide factor that is critical to the successful application of the processor, it gets the user's inputs in the Requirements Definition step of the processor development -- where they will most effectively impact the design of the processor.

We intend to use the wizard system in the Requirements Definition step of the processor development. The software for the wizard system presently exists so the system can be used without any cost for its development.

Design

Since the major portion of the processor to be developed consists of the user interface (in contrast to the development of processors involving complex data structures or computational routines), we suggest using Ken Orr's (1981) Structure Requirement Definition methodology of identifying the outputs required from the processor and then, systematically in sequence, identifying the processor and then, systematically in sequence, identifying the processor functions needed to produce those outputs - call those functions level one functions. This process is continued by identifying each successive level of processor functions (i.e., level two functions provide the inputs needed by level one functions etc.) until the primary inputs, the user keyboard inputs and data files, are identified.

Since, the processor uses only a few inputs from the CD and expert, the majority of primary inputs will be from data files. This design methodology is easy to use and understand, and uses documentation which is self explanatory.

As mentioned previously, many of the processor's algorithms are already part of the MAP processor of Connelly (1986). Thus much of the design work will involve the identification of the data for the processor's several data bases. The source and structure for the data bases is described in the section of the paper entitled "Types and Location of Data Required for the Processor."

Another large task of the design stage will be the development of the unit MOE. The process for producing the unit MOE is described in the following section.

The Development of Unit MOE

The source of data on unit effectiveness is the effectiveness preference of the designated authority above the unit. A rule accurately representing the authority's judgements (MOE) must be carefully established: its dependent variables must be tested to insure they are correct and are complete; weigthing of its variables must be tested by having the authority carefully examine their implications on the unit effectiveness scores that are assigned by the MOE for likely mission outcomes; and the sensitivity of the rule (MOE) must be specified so that important changes in rule variables result in a correct and significant change in the effectiveness score produced by the rule (MOE).

The procedure for developing the unit MOE is as follows:

1. The experts (authorities) are asked to list the factors (such as "selection of proper route") they use when they assess the effectiveness of a unit.

2. The experts are asked to identify the variables they use to evaluate each of the factors they identified in step one. For instance one variable used to evaluate "selection of proper route" might be "potential speed of vehicle along route" or "% of route concealed from enemy."

3. The experts to identify particular values for each of the variables. For instance, one value might be "30 m/h for a clear day" for the variable "potential speed of vehicle along route." Another might be "50% of the route is concealed from the enemy" for the variable "% of route concealed from enemy."

4. For each of the factors identified in step 1, the processor forms combinations of values for the variables used to evaluate that

factor. Each combination of the variable values represents a particular situation that could occur during the mission. For instance, the combination "30 m/h" combined with "50% of the route concealed from the enemy" could be one combination of variable values describing the quality of the route selected.

5. The experts are asked to provide a score from one to 100 evaluating the degree of intensity or strength of each combination of variable values.

6. The processor then forms combinations of the factor values.

7. The experts are asked to provide a score evaluating the unit effectiveness for each combination of factor values.

8. The processor uses the scores provided by the experts to synthesize rules for evaluating each factor and for assessing the unit effectiveness in accomplishing its assigned mission; i.e., the unit MOE.

It is desirable that the designated authority consist of more than one individual so the combined consideration of all are used to synthesize a unit MOE. But there must be a method for resolving differences among individuals because such differences may indicate either a defect in an interim specification of the MOE or alternatively may indicate a different unit effectiveness preference among the designated authorities. A simple method is used to do this. It is as follows: the authorities consist of three experts. One of the experts is designated as the senior authority, usually the senior in command. The other experts provide their MOE data in turn with the second given the results from the first. The second expert can modify the data from the first but both sets (i.e., the data from both the first and also from the second) are passed on to the senior expert. Each expert has the advantage of seeing the data provided by the predecessors and can agree with it or provide modified data. The final expert, the senior expert, reviews data from all the other experts and produces the final opinion - the final identification of factors, variables used to measure those factors and evaluation scores. The rules for evaluation the factors and the unit MOE are synthesized from only the senior expert's data.

Questionnaire for the Expert. The first part of the questionnaire asks the expert the following.

1) Enter the factors that they use when they assess effectiveness of a unit.

2) For each factor, enter the variables they use to measure that factor (note, variables must be quantifiable such as: number of rounds, distance traveled, time to complete).

3) Consider a specific unit they are familiar with and for that unit identify and enter values for each variable -- these values are called the "baseline" values.

4) Establish and enter for each variable, the smallest amount of increase and the smallest decrease from the baseline value that has a significant impact on unit effectiveness. These values are used as described subsequently to specify the sensitivity of the unit MOE.

5) Enter any and all assumptions used to provide any of the above information. Indicate the factors or combination of factors involved in the assumption and then the conditions assumed to exist.

Note that this part of the questionnaire obtains the factors the experts use to assess unit effectiveness but at that time the factors are not quantified. Consequently, the questionnaire asks: "What variables (which must be quantified variables) do you use to measure each factor?" For instance, if a unit factor is identified as: "select proper route," then the variables identified by the experts for measuring the quality of route selection might be:

1. Potential speed along route;
2. Percent of route concealed from enemy;
3. Maximum distance of route from center of assigned sector;
4. Percent of route in which assigned sector can be scanned.

Each of these variables are quantitative variables. Suppose that the values for these variables are given by the expert as:

	Baseline Value	Largest Value	Smallest Value
1. Potential speed along route:	20m/h	30m/h	10m/h
2. Percent of route concealed from enemy:	80%	90%	70%
3. Maximum distance of route from center of assigned sector:	100m	200m	50m
4. Percent of route in which assigned sector can be scanned:	90%	100%	80%

The purpose of part two of the questionnaire is to establish the regression equation the expert uses to evaluate each factor. This not only provides the rule but also provides a way of quantifying each factor; i.e., the number assigned to each factor has specific meaning via the mathematical function and variables. The purpose of part three of the questionnaire is similar to that of part two except that part three is used to establish the rule the expert uses to assess the effectiveness of the unit performing the assigned mission function.

Part two of the questionnaire consists of a matrix of combinations of variables values. The expert is asked to score each combination, using any scale he chooses. For example, given the variables and variable values indicated above for the factor "potential speed along the route," the matrix of value combinations would be:

	---Variable Values---				Factor Score
Combination	#1	#2	#3	#4	for Combination
1) Baseline	20	80	100	90	
2) High 1st variable	30	80	100	90	
3) High 2nd variable	20	90	100	90	
4) High 3rd variable	20	80	200	90	
5) High 4th variable	20	80	100	100	
6) Low 1st variable	10	80	100	90	
7) Low 2nd variable	20	70	100	90	
8) Low 3rd variable	20	80	50	90	
9) Low 4th variable	20	80	100	80	
10) All high	30	90	200	100	
11) All low	10	70	50	80	

Note that there is a combination where each variable has three values (baseline, high, low) and the other variables have the baseline values. The combinations of all high and all low give the experts the opportunity to establish the max and min values of their scales. Experts complete this part of the questionnaire for each of the factors they identify.

When the experts complete the questionnaire by entering a score for each combination, the scored set of variable combinations constitute a specification for the rule for scoring the associated factor. The method for synthesizing that rule is given in a subsequent section.

Turning now to part three of the questionnaire, a matrix of combinations of factor values is formed much the same way as the matrix of variable values, described above, was formed. For instance, if four factors were identified by the experts as those used for assessing unit effectiveness, then the matrix would be:

	Factor	Factor	Factor	Factor	Unit Effectiveness Score for
	#1	#2	#3	#4	Combination
Baseline	8	8	8	8	
Factor 1 high	9	8	8	8	
Factor 2 high	8	9	8	8	
Factor 3 high	8	8	9	8	
Factor 4 high	8	8	8	9	
Factor 1 low	7	8	8	8	
Factor 2 low	8	7	8	8	
Factor 3 low	8	8	7	8	
Factor 4 low	8	8	8	7	
All factors high	9	9	9	9	
All factors low	7	7	7	7	

Note that the experts are asked to specify a value for the minimun acceptable level of unit effectiveness. The minimun acceptable level is the unit effectiveness score that must be equaled or exceeded for the unit's effectiveness to be considered satisfactory -- a criterion of acceptability.

Interact with each Expert. As the experts complete parts two and three of the questionnaire, the expert can become unsure of his rule. Thus, feedback, in the form of examples sorted on the effectiveness score and graphs of effectiveness score vs. factor values, are used to insure that the score given to each combination are what the expert actually intends. The principle here is that whenever judgment data are collected from experts, a substantial reduction in errors of commission occurs if the data collected are transformed into a different form and then feed back to the expert for review, providing "another viewpoint" (Connelly, 1982).

The uni MOE synthesis procedure builds an accurate model of the rule the experts use to assess the effectiveness of a unit accomplishing the assigned mission. The procedure is to test a fit of a linear function of the factors (independent variables) first, using a regression analysis. Test of the fit of the linear function compares the scores produced by the regression equation to the scores given by the expert for each combination of variables and factors in part two and part three f the questionnaire, respectively.

Programming and Testing

The majority of the program coding will be assigned to the support staff. These programs will be modules with well specified inputs and outputs. The main program which calls each module will be under the control of the core project team. Consequently, the support staff will conduct the module test while the core team will conduct the integration and system tests.

Installation

Installation involves testing to determine if the processor will work when entered in the equipment the user has available, and if the user can actually use the processor properly. The first test is to verfiy that the processor will run on all the equipment configurations identified in the initial surveys conducted during the requirements definition phase. Next, the ease of use will be verified by testing the ability of a sampling of users (CDs and experts) to operate the processor properly.

TYPES AND LOCATION OF DATA REQUIRED FOR THE PROCESSOR

The processor performs four functions: it develops objectives and system performance criteria; identifies conditions which may impact system performance; and it can be used to develop new unit MOE. The types of data required for each of these functions is described in the following paragraphs as is the locations of the types of data and how to obtain them.

To Develop Objectives

Type of Data	Location of Data Type
1) Type of fielding unit	1) MAA; Concept papers; O&O plan
2) Deficiencies	2) MAA; Concept papers; O&O plan
3) Mission of Unit	3) MAA; Concept papers; O&O plan
4) Functions and subfunctions	4) The TRADOC schools developed a set of tasks and subtasks for each which mission area. The combat developers use these to write MAAs. They have been collated and are in Appendix A. However, other sets of already developed functions and subfunctions are available. SAIC is using one such set for the SORD system which will be used by TRADOC combat developers to develop the organizational structures for units. Also, the Dynamics Research Corp. is developing a revised set of functions and subfunctions for TRADOC use. This is being done on an ARI contract whose COTR is Dr. David Promisel.
5) Activities; activity specifications	5) These were used in section 3.2.2 to further specify the objectives of the unit. These were developed by SAIC for the SORD system. Others could be used in their stead if they are found to have more user acceptance. The others include those being developed by the Dynamics Research Corp.

To Develop System Performance Criteria

Type of Data	Location of Data Type
1) Candidate lists of system characteristics baseline values to form basis of system performance criteria	1) ROCs of predecessor and related systems
2) Concept of Ops.; Larger force description; Logistics of unit; Enemy forces; Probable locations unit will be used;	2) MAA; O&O plan

To Identify Conditions

Type of Data	Location of Data Type
1) List of candidate operational, tactical and environmental conditions	1) MAA, O&O, ARTEPS

To Develop Unit MOE

Type of Data	Location of Data Type
1) Factors to form the basis of unit MOE	1) Judgements of experts at TRADOC schools; Also, each MAA, TOE and O&O plan has the factors of a unit MOE

Locations

How to obtain each of these types of documents and the locations of the data within them are described in the following paragraphs.

The MAA

The MAA for each mission area typically resides in two locations. It is usually in the files of the Studies and Analysis branch of the Combat Developments division of the proponent school and at DCSOPS. It can be obtained from either location with the proper clearances.

MAAs typically describe the threat, doctrine, battlefield tasks and subtasks, EEA, MOE and the deficiency. However, because MAAs are usually a compilation of several studies and do not rigorously adhere to a prescribed format, it is not possible to specify the precise location of each of the types of data within an MAA.

O&O Plan

The O&O Plan for each unit also typically resides in three locations. It is usually, like the MAA, in the Studies and Analysis branch of the proponent school. It is also usually in the office of the TRADOC Staff Systems Officer and at DCSOPS.

The following format, used when preparing an O&O Plan, can also be used as a guide to finding the types of information the combat developer will need to retrieve from an O&O Plan. The descriptions accompanying the format indicate the types of information that should be contained in each paragraph of an O&O Plan. The very first section of an O&O Plan should specify the location in the MAA where the deficiency is identified.

I. Purpose — This describes the need for an operational capability to defeat the threat and eliminate an operational deficiency. It states where in the MAA the deficiency is identified and how the need was develop from the described deficiency. (The need is stated in broad characteristics only; e.g., a capability is needed to defeat enemy armor at "X" kilometers.)

II. Threat — This describes the threat to be countered and the operational deficiency to be eliminated.

III. Operational — This describes how, what, when, and where the system will be employed on the battlefield and how it will interface with other systems (attach Operational Mode Summary/Mission Profile as an annex). Communications support requirements also are addressed.

IV. Organizational — This discusses the type of units that will employ and support the system and, when appropriate, the system to be replaced. (When the system is decided on, the number of systems estimated to be provided to each type of unit is included.) This plan will support preparation of the Basis of Issue Plan (BOIP), the Integrated Logistic Support Plan (ILSP) and the identification of key ancillary items.

V. Personnel Impact — The design of the system should consider personnel skills available to operate and maintain the system. Generation of new MOSs should be avoided where possible. (When the system is decided on this section includes an estimate of the number of people and skills estimated to operate and maintain the equipment, by type unit.) This plan will support preparation of the Qualitative and Quantitative Personnel Requirements Information (QQPRI), the Personnel Support Plan, and assist in the LSA process. It also addresses Manpower/Personnel Intergration (MANPRINT) issues.

VI. Training Impact

The design of the equipment should consider type and extent of training required. (When the system is decided on, the type and amount of training devices and simulators should be described.) This part of the plan will support preparation of the Training Support Plan.

VII. Logistics Impact

The system must be supportable by the Standard Army Logistics System and use standard tools and Test, Measurement, and Diagnostic Equipment (TMDE). (When the system is decided on, the proposed levels of maintenance, support concept, TMDE, Automatic Test Equipment (ATE), and Built-in Test Equipment (BITE) concepts should be included.) This part of the plan will support preparation of the ILSP.

TOE

The combat developer may need to use the TOE as a reference tool in gaining a TOE for each unit may typically be found in the same three locations at which the O&O Plan can be found. TOE are the basic documents that describe a type unit. They provide information on a unit's mission, capabilities, limitations and describe in detail the minimum essential personnel and equipment to accomplish wartime missions. The function of TOE can be stated as: 1) providing information on requirements (but not authorizations), 2) describing minimum essential requirements, 3) serving as the Army's base organization document, and 4) serving as a model for structuring Modification TOE (MTOE). A TOE is organized into the following sections and sub-sections:

Section I

- Heading
- Organization Chart
- Mission
- Assignment
- Capabilities
- Basis of allocation
- Category
- Mobility
- Doctrine

Section II

- Personnel and equipment, by paragraph
- Recapitulation
- Remarks

ARTEPS

ARTEPS are available on each extant unit. These contain in several sections a list of conditions that the unit should train under. The conditions include the environmental, tactical and operational types.

INSURING ACCEPTABILITY AND USABILITY

Despite the proliferation of computers in civilian and military organizations, acceptance and exploitation of the machines by their intended users is still far from optimum. Numerous examples exist of computer systems that are designed to meet high expectations, but which remain underused throughout their costly life.

User acceptance of computer systems is a frequently neglected aspect of software evaluation. User acceptance is a function of ease of use, perceived usefulness of output, and validity of output. The software developer typically evaluates product effectiveness in terms of speed, accuracy, quality of output, and the extent to which it improves human performance. However, users reject even highly effective software systems for any number of reasons. Two important reasons are ease of use and perceived usefulness. Therefore, the software developer alone cannot judge the effectiveness of the product. Users must also judge the product as easy to use and useful.

Ease of Use

The general notion of user acceptance includes both ease of use and perceived usefulness. Another way to view this issue is in terms of reliability of use and validity of output. If a software system does not have both these attributes, then it will not be accepted by the user. Four areas will be considered in evaluating ease of use: the skill levels of potential users; the type and specificity of feedback given to users; the consistency between what users request and what they receive; and memory demands the software system places on users (Liffick, 1985).

Operator Skill Evaluation

The user interface will be designed to match the skill of the users. The interface will provide embedded training for the novice user. The information the novice user must have to make a decision must be known and available. Experienced users may need less information in order to use the system. Given the newness of this software system, it can be assummed that users will be novices. Therefore, it will be necessary to create a dynamic system, one that changes as the user becomes familiar with it. The processor will include separate tracks for different user experience levels.

The point at which a novice user becomes an experienced user is not easy to define. It is usually not the case that one day the user is a novice, and the next he or she is experienced. Even an experienced user might want to use a feature he or she has not tried before, and regress briefly to the novice stage. The processor will allow an experienced user to function as a novice, on demand, then return to the experienced user mode. Switching from experienced to novice mode will be simple, and the user will clearly know where he or she is in the system.

Information to the User

Menus and feedback provide the information the user needs to navigate through the system. Systems described as user friendly are usually menu oriented. Feedback, no matter how simple, is important to keep the user informed about every action that has been requested. Feedback lets the user know what the system is doing, so the user knows that what has been requested has been accomplished. Given the many suspicions that novice users tend to have about computers, this is important. All feedback should be positive. When the user has done something incorrect, the system will clearly identify the incorrect action as well as a direction about how to continue. This keeps the user from having to guess what to do next.

Consistency

User effectiveness is increased where there is consistency in rules, and little ambiguity. Ambiguity requires the user either to make a decision with incomplete information, or waste time searching the documentation for a resolution to the ambiguity. Therefore, consistent procedures will be established for user interactions. The consistent use of rules will allow the user to make assumptions about how things work within the system.

User Memory Demands

It is important to minimize the demands on human memory. A help function for the user is the ideal way to limit memory requirements. Such a function can usually be entered at any time by the user. The help function provides details about how each part of the system works, what the various commands of the system are, and what what the formats for inputs are. If the user needs more information than is provided in the help function, he or she will also have the option of entering novice mode. As mentioned above, the user will be able to return to experienced used mode when the additional help is no longer needed.

Summary

Methods to ensure ease of use are: analysis of potential users; feedback about what the system is doing and where you are in it; consistency of rules so that assumptions can be made about how things work in the system; and providing help to limit memory demand on the user. In considering these four areas, a brief target audience analysis will be conducted. The products of this analysis will not only increase the ease of use for the eventual user, but also ensure that the user is an active participant in the design process.

Perceived Usefulness

Participation in product design by the user may well lead to a match between what the software developer sees as effective and what the user sees as useful. User acceptance is a combination of reliability, ease of use, and validity and perceived usefulness of output. No single one of those is sufficient for user acceptance. For example, the user may find the processor easy to use, but of no particular value. In that case, user acceptance is low. In contrast, the user may find the product difficult to use but of great value; the user may struggle to use the product, but user acceptance will be low.

The concept of product usefulness may be measured subjectively and objectively. Subjective measures evaluate the attitude of the user toward the product; e.g., is the product helpful? is it difficult to use? does it seem to be effective? Objective measurments can also be taken. Variables to be measured objectively should be directly related to the user's job and measurable by the software developer; for example, frequency of use, length of session, use of output, and improved human performance. By selecting job relevant dimensions to measure, there is a good chance that the effectiveness sought by the software developer will closely match the perceived usefulness of the processor by the user.

Causes and Results of Poor User Acceptance

Even when the performance of software design is excellent, the problem remains of how to encourage its use in the field (Donnell, Fineberg, and Carter, 1987). Procedures for improving implementation and use require an understanding of the user's attitudes and perceptions toward the product and its use. The user's background and experience with computers affect user acceptance. The fit of the product within the context of the existing job situation will affect user acceptance. If the user detects conflicts between product use and existing doctrine, acceptance will be poor. Finally, product performance will affect user acceptance. The product must run reliably with little downtime and product outputs that are correct.

Some of the specific problems listed in Donnell et al. (1987) that may cause poor user acceptance include:

- Lack of user confidence, reflecting perceived unreliability, often resulting from failures, errors, or breakdowns in the sensitive early stages of system introduction.

- Divergence from perceived function, where the hardware or software manifestation of the system is at odds with the user's idea of what it does or should do.

- Divergence from individual needs, where the user feels that his or her specific requirements, preferences, tastes, etc., are ignored or even offended by specific system characteristics.

- Divergence from individuality, where the user feels unable to influence the system personally.

- Threat to privacy, where the user feels he or she is liable to some form of exposure (data or decisions) as a result of system use.

- Threat to security or self-esteem. Of particular importance to acceptance, this often reflects the reluctance of well-placed users to make themselves look foolish by failing to master seemingly complex new technology. It may also reflect a personal conclusion that one's job is vulnerable to computer encroachment; or, alternatively, that computer use diminishes the status of that job by incorporating menial elements.

Poor user acceptance results in a variety of user responses. A list of common responses is presented in Figure 6. If user acceptance problems are discovered before the system is fielded, solutions will be easy to implement. One of the major goals of product one is to avoid user acceptance problems early in the development process. To do this it is essential that measuring user acceptance be an integral part of all product one development efforts.

Assessment of User Acceptance

User acceptance of product one will be measured objectively and subjectively. A subjective assessment indicates how satisfied the user is with the system, and is accomplished through user interview and questionnaire responses. An objective assessment indicates how much the user uses the system, and is made by monitoring actual system use.

Subjective Measures

Subjective data concerning user accepance can be obtained via structured interviews or questionnaires. User acceptance and product usefulness are the two broad categories used in subjective evaluation (Donnell et al., 1987). Users will be asked to rate Product 1 on the following dimensions of user acceptance:

1. The system is matched to the user.

2. The system provides the critical variables needed to solve the problem.

3. This product can handle simple and complex functional capability problems.

4. The product does not add to the already considerable information overload within the operational and organization planning effort.

5. Use of the product will not require more expertise from the typical user than is likely to be available in the operational environment.

6. People can easily under the procedures to be followed in using product one.

7. The product provides a common language, facilitating easy communications between members of the decision making team.

Response	Definition	Comments
Dis-Use	Reliance on other information sources.	Requires existence of alternative information source, user with sufficient discretion.
Mis-Use	"Bending the rules" to short-cut operational difficulties.	Requires significant knowledge of system. May negatively impact system integrity.
Partial Use	Use of (perhaps inappropriate) subset of system capabilities.	Users frequently adopt "satisfying" strategy, may not learn most relevant system capabilities.
Distant Use	Interposition of operator between user and system.	Requires high status and discretion. Typical response of managers.
Modification of Task	Changing the task to match capabilities of system.	Prevalent when tools are rigid, problem is unstructured, as in scientific problem solving.
Compensatory User Activity	Compensation for system inadequacies by additional user actions.	Typical with users of low discretion, as clerks.
Direct Programming	Programming by user, in order to modify system capabilities to suit needs.	Typical response of computer-sophisticated user, as scientific and engineering user.
Frustration and Apathy	Response of user when above actions are inadequate or unsatisfactory.	Involves lack of user acceptance, high error rates, poor performance.

Figure 6. User Responses to Inadequate System (from Ramsey and Atwood, 1979).

8. The product contributes to the essential flow of intergroup information, or communications, necessary for effective decision making.

9. The use of the product is consistent with, and would not interfere with O & O planning.

10. A user can be confident in product one decisions.

11. If implemented in an operational environment, use of the product one can be expected to increase as time progresses.

12. The use of product one in an operational environment is a realistic goal for the near future.

Product one users will also rate the product on the following dimensions of product effectiveness:

1. Enables sufficiently rapid and complete responses to aid the needed system capability decision-making process;

2. Encourages the user to explicitly identify relevant objectives and to prioritize them;

3. Encourages effective response to the issues most relevant to determination of system capabilities;

4. User can readily prepare data, input data, and extract understandable results;

5. Encourages the decision maker to consider a wide range of options or possible system alternatives;

6. Encourages one to think critically and realistically about problems and prospects for implementation of the selected decision;

7. Focuses and enhances appropriate and constructive decision maker discussion concerning various system capabilities under consideration;

8. Possesses considerable generality so that many different problems can be relatively easily accommodated;

9. The value of the product will increase as the complexity of problems to which it is applied increases.

Objective Measures

Objective measures of user acceptance, such as the duration and frequency of user sessions, and the time taken to generate a training constraint, should also be studied. This will be done in as unobtrusive manner as possible, with no demands placed on the user's time.

Summary

An effective human performance aid must be used. We will measure Product I user acceptance, subjectively and objectively. Subjective measures will assess how well the user "likes" the system, and whether the user indicates he or she would use the system in the future. Objective measures will consist of reports of actual system use, and an assessment of the quality or correctness of output.

TRAINING OF USERS

Two components will be built into the processor to enable users to learn to use its capabilities. The first is a comprehensive embedded training (ET) capability and the second is a context-sensitive Help and Explanation capability. It is anticipated that the two components will share many data elements and software routines, since their purposes and functions are similar.

Embedded Training Capability

The ET capability will be accessed from the operating system level. A unique command will be provided to call up and begin the ET component, separate from normal processor functions. This capability allows "off-line" training to prepare new users to learn its functions and capabilities, as well as review or sustainment for more experienced users. The ET component will contain the following functional capabilities:

- Modular Lessons: Specific topics will be organized into lessons which can be used independently. An overall structure will guide initial training, but the user will not be constrained to use the training modules in any specific order. The following major lesson topics are anticipated:

    - Introduction
    - Developing System Objectives
    - Developing System Criteria
    - Identifying Conditions Affecting System Performance
    - Sensitivity Analysis/Tradeoffs
    - How the Processor Works
    - Modifying Data Bases
    - Developing new unit MOEs

    The first four modules are designed to enable the first-time user to utilize the processor to derive system requirements. The last four deal with advanced topics for more experienced or interested users, or those who need to use the more advanced capabilities of the processor.

- Guided Practice and Worked Examples: Much of the training provided by the ET component will consist of hands-on exercises with extensive guidance for the user. Exercises will concentrate on accomplishing specific steps of using the processor, and will contain error diagnostics.

- A Balanced Mix of Knowledge and Hands-On Training: Some users will be uninterested in "how the product works," and will wish to emphasize practical capabilities. Others will develop an interest in how the product does what it does to produce its outputs. The content and structure of training will accommodate both extremes, as well as many intermediate points on the "theory-practice" continuum.

- Checkpoint and Resume: The processor's users are busy people with many demands on their time. Thus the user will not be asked to dedicate time to complete even a single module of training at one session. Each user's progress in training will be monitored by a control feature in the ET component, and a user will be able to suspend training at any point and resume from the same point at a later time.

Context-Sensitive Help and Explanation Facility

This facility will enable the user to request help and explanations at any time the user is actually interacting with the processor. Context sensitivity of this feature refers to the fact that the processor will have information about what the user is attempting to accomplish during any interaction. Using this information, the processor will provide guidance and explanations of how to accomplish the particular function. The processor will present information regarding why particular inputs, judgments, etc. are needed to accomplish the interaction. Guidance will always be provided when the user invokes the help capability. "Why" information will only be presented at the explicit request of the user.

The user interface with the help and explanation capability will be provided through a "hot-key" approach, with one function key (or the equivalent) always set aside to request help. If the information contained in the help or explanation requires more than one full display screen to present, the user will utilize the normal up and down cursor-control or scrolling keys to move forward or backward through the information presented. If there are options, choices, or responses associated with a help or explanation display, the user will be presented with a "pull-down" menu of choices, above the normal display area for help information. Choices will be made by moving a block cursor to the desired option or response (using the left and right cursor control keys) and using an "enter" or execute key to invoke the choice desired.

INSTITUTIONALIZING THE PRODUCT

There are two approaches that will be used to institutionalize the product. One is "campaign" actions that will be taken and the other is to build into the product characteristics that will help insure its institutionalization.

Actions to be Taken to Institutionalize

The institutionalization plan includes four campaign action items. Chronologically first will be getting the potential users of the product involved in all stages of its development. The potential users will be involved in the design and testing stages through both informal commenting and more formal pilot-testing. A frequent dialogue will be established with the more cooperative potential users. They will be asked for input on data base items and structures, and interface design in addition to the more basic procedures of the product.

Potential users include the combat developers in the trenches at each of the schools. In addition, their superiors up to the typical heads of the combat developments directorates will be encouraged to be involved in the development of the system. Similarly, appropriate personnel at headquarters TRADOC also will be involved. Finally, AMC personnel also will be encouraged to participate.

The reasons for the involvement of the potential users are:

1. To develop a critical mass of positive potential users prior to the availability of the product. They will help institutionalize the product by using it and promoting it.

2. To obtain data from the potential users to help tailor the product more to their needs and desires. This will make it a more attractive product and thus help insure its institutionalization.

3. To be able to say we consulted with the users in the development of the product and that we have their support and approval.

The second campaign action item will be to obtain the support of a general officer. This will be accomplished after the product is partially developed so that its rudiments can be shown and demonstrated to help obtain support. Also, the interaction and support of the users will be marshalled to gain the support of a general officer. In addition, briefings and white papers will be used hailing the efficiency, cost and other benefits the product will yield.

The third campaign action item is to have the use of the product included in the course taught to combat developers at Ft. Leavenworth. Allen Corporation of America teaches the course to combat developers. Since Allen Corporation is part of the ASA team producing this product, it will be possible to have Allen include a new piece in the course specifically devoted to promoting product. In addition, besides promoting the product, the course will teach the combat developers how to use it and make them familiar and comfortable with it. All of these additions to the course will help institutionalize the product.

Inherent Features Fostering Institutionalization

There are six features which will be inherent characteristics of the product or results it will yield. All six will foster its institutionalization. The six are:

- Face validity
- Reduces labor
- Lowers cost of combat developer's process
- Lowers cost of system development
- Produces a formatted audit trail
- Helps develop O&O plans

Through interaction with the users during the development and testing of the product will come face validity. Face validity will go a long way toward institutionalizing the product. The face validity will partially result from the users' feedback. Such feedback will provide guidance for the the design and development of the product. Thus it will have the look and feel of the users.

Obviously the product will result in a reduction of the amount of labor required by combat developers to produce system requirements. This feature of the processor will almost in and of itself cause it to be institutionalized. Many system development efforts have resulted in elegant systems that were never used. Often this was because the systems required the users to do more than they did before. On the other hand, everyone appreciates a job aid that actually makes their job easier especially if it is because they have to do less. The reduced labor from the combat developers also will result in their requirements generation process costing less.

Similarly, the processor will lead to a lower cost for the development of systems than was previously experienced. This will be the result of having specific criteria with which to judge the performance of the system. Such criteria will strongly encourage the development of a system designed to perform its mission. Thus will be avoided any costly redesign efforts prior to developmental testing and any retro-fitting after operational testing. In addition, designing the system to meet specific performance criteria also will help the system avoid having to undergo the typical product improvement efforts after fielding. All of these reduced cost aspects will be effective in obtaining the support of a general officer.

The processor will result in a formatted audit trail which will document each system requirements development effort. This will be an especially attractive feature of the processor for the present requirements generation process often leads to unanswerable questions about decisions and system requirements. Part of the reason for the present process resulting in unanswerable questions is that the process is not proceduralized and thus requires more of an audit trail than a proceduralized one. Moreover, the present process provides no format or easy to use vehicle for generating an audit trail.

Finally, the processor will help combat developers produce an O&O plan. Again, any time a new product makes someone's job easier the product has a high probablity of being used and thus being institutionalized.

REFERENCES

Bendig, A.W. (1953a). The reliability of self-ratings as a function of the amount of verbal anchoring and the number of categories on the scale. Journal of Applied Psychology, 37, 38-41.

Bendig, A.W. (1953b). Reliability of short rating scales and the heterogeneity of the rated stimuli. Journal of Applied Psychology, 1953, 38, 167-170.

Bendig, A.W. (1952a). A statistical report on a revision of the Miami instructor rating sheet. Journal of Educational Psychology, 1952, 43, 423-429.

Bendig, A.W. (1952b). The use of student rating scales in the evaluation of instructors in introductory psychology. Journal of Educational Psychology, 1952, 43, 167-175.

Brooks, F.P. (1975). The mythical man-month. Philippines: Addison-Wesley Publishing Company.

Campbell, J.P., Dunnette, M.D., Lawler, E.E. III, and Weick, K.E. Jr. (1970). Managerial behavior, performance and effectivenss. NY: McGraw-Hill.

Connelly, E.M. (1986). User handbook for MAP computer programs. Reston, VA: Performance Measurement Associates, Inc.

Connelly, E.M. (1982). A comparisor of the accuracy and completeness of problem solutions produced by example-solutions and program code. Reston, VA: Performance Measurement Associates, Inc.

Connelly, E.M. (1981). Development of system performance measures. Vienna, VA: Performance Measurement Associates, Inc.

Connelly, E.M., Zeskind, R.M., & Chudd, G.P. (1977). Development of a continuous performance measure for manual control. (Final Report on Contract F 3365-75-C-5088, AMRL-TR-76-24).

Darlington, R. B. (1968). Multiple regression in psychological research and practice. Psychological Bulletin, 6, 161-182.

Dawes, R. M. and Corrigan, B. (1974). Linear models in decision making. Psychological Bulletin, 81, 95-106.

Donnell, M.L., Fineberg, M.L., and Carter, C.F. (1987). Research and applications to enhance human performance in command control systems. (Draft interim technical report, number 87-XX). McLean, VA: Science Applications International Corporation.

Federal Information Processing Standards, National Bureau of Standards. (1983, June). Guidelines for lifecycle validation, verification, and testing of computer software. (Federal Information Processing Standards Publication FIPS Pub 101). Washington, DC.

Finn, R. H. (1972). Effects of some variations in rating scale characteristics on the means and reliabilities of ratings. Educational and Psychological Measurement, 32, 255-265.

Flanagan, J. C. (1954). The critical incident technique. Psychological Bulletin, 51, 327-358.

Goldberg, L.R. (1970). Man versus model of man: A rationale, plus some evidence for a method of improving on clinical inferences. Psychological Bulletin, 73, 422-432.

Guion, R.M. (1961). Criterion measurement and personnel judgments. Personnel Psychology, 14, 141-149.

Guirreri, J.A. (1986). Catalog of Wargaming and Military Simulation Models. Washington, DC: Joint Analysis Directorate, OJCS.

Hoffman, P.J., Slovic, P. and Rorer, L.G. (1968). An assessment of configural cue utilization in clinical judgment. Psychological Bulletin, 69, 338-349.

Kendal, W.D. (1955). Industrial psychology. Annual Review of Psychology, 6, 217-250.

Keyes, J.D. (1980). CASTEFOREM III, An Overview. Proceedings of the Nineteenth Annual US Army Operation Research Symposium. Fort Lee, VA. October, 578-598.

Landy, F.J. and Farr, J.L. (1980). Performance rating. Psychological Bulletin, 87, 72-107.

Lapidus, L. and Luus, R. (1967). Optimal control of engineering processes. Waltham, MA: Blasidell.

Liffick, B.W. (1985). The software developer's sourcebook. Reading, MA: Addison-Wesley Publishing Company, Inc.

Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. Psychological Review, 63, 81-97.

Nagle, B.F. (1953). Criterion development. Personnel Pshchology, 6, 271-289.

Nunnally, J.C. (1978). Psychometric theory (2nd ed.). NY: McGraw-Hill.

Orr, K. (1981). Structured Requirements Definition. Topeka, KS: Ken Orr and Associates, Inc.

Pontryagin, L.S., Poltyanskii, V.G., Gambrelidze, R.Z., and Miskchenko, E.F. (1962). The mathematical theory of optimal processes. NY: Interscience Publishing.

Ramsey, H.R., and Atwood, M.E. (1979). Human factors in computer systems: A review of the literature. (Report SAI-79-111-DEN). Englewood, CO: Science Applications International Corporation.

Slovic, P. and Lichtenstein, S. (1971). Comparison of Bayesian and regression approaches to the study of information processing in judgment. Organizational Behavior and Human Performance, 6, 649-744.

Smith, P.C. (1978). Behaviors, results, and organizational effectiveness: The problem of criteria. In M.D. Dunnette (Ed.), Handbook of industrial and organizational psychology. NY: Wiley.

Thomas, G. S. and Cocklin, T. G. (1983). A model of mission accomplishment in simulated battle. Technical Report 599. U.S. Army Research Institute for the Behavioral and Social Sciences, Alexandria, VA. (AD A148 234)

APPENDIX A

STATEMENT OF THE PROBLEM

STATEMENT OF THE PROBLEM

This appendix describes the problem and the context that generated the requirements for product one. The following subsections describe the current materiel acquisition process (MAP) and the events that set the MAP in process. Within the description of the MAP is a description of the process of developing objectives and criteria for a new system and the context in which these are developed. Also described are the most common problems associated with new system criteria and finally the probable user of the processor. This section concludes with a detailed description of the requirements for product one. The requirements are based on the essentials of the four subsections which preceed them.

The Context - The Army Materiel Acquisition Process (MAP)

Prior to the initiation of the MAP, several events must occur. Most of these are part of the Army's concept based requirements system. Primary among the events is the development of a mission area analysis (MAA) identifying a deficiency in the Army's ability to accomplish its mission.

After a deficiency is identified and a materiel solution agreed upon as the most appropriate solution to the deficiency, the MAP usually begins. The development of new systems is one of the types of materiel acquisition which is part of the Army's MAP. For the development of new materiel systems, the first stage of development is the most important stage relative to the purpose of this paper. The first stage is the concept exploration stage. In this stage the early performance requirements for a new system must be developed. Thus it is in the concept exploration stage that combat developers should begin to use the processor.

Events Leading to the Initiation of the MAP

The Concept Based Requirements System (CBRS) is the basis from which all Army requirements evolve. The major parts of the CBRS are the development and analysis of Functional Operational Concepts (FOCs), the MAA, the Battlefield Development Plan (BDP), the Mission Area Development Plan (MADP) and the Operational and Organizational (O&O) Plan. The major parts and flow of the CBRS process are shown in Figure 7. The CBRS begins with the consideration of the Army mission, historical perspective, threat and technological forecasts. These four areas are examined and umbrella concepts are developed which define, in general, how the Army will fight now and in the future.

The umbrella concepts lead to the development of the more specific FOCs. Approved FOCs are integrated into the MAA process. MAAs include the analysis of operational concepts, the determination of broad functions, and the identification of deficiencies in the areas of doctrine, training, organizations and materiel. The results of the MAA are a list of deficiencies and a series of actions required to correct the deficiencies. However, the MAAs do not contain performance objectives or criteria for new systems. TRADOC tasks proponent schools and centers to develop the MAAs

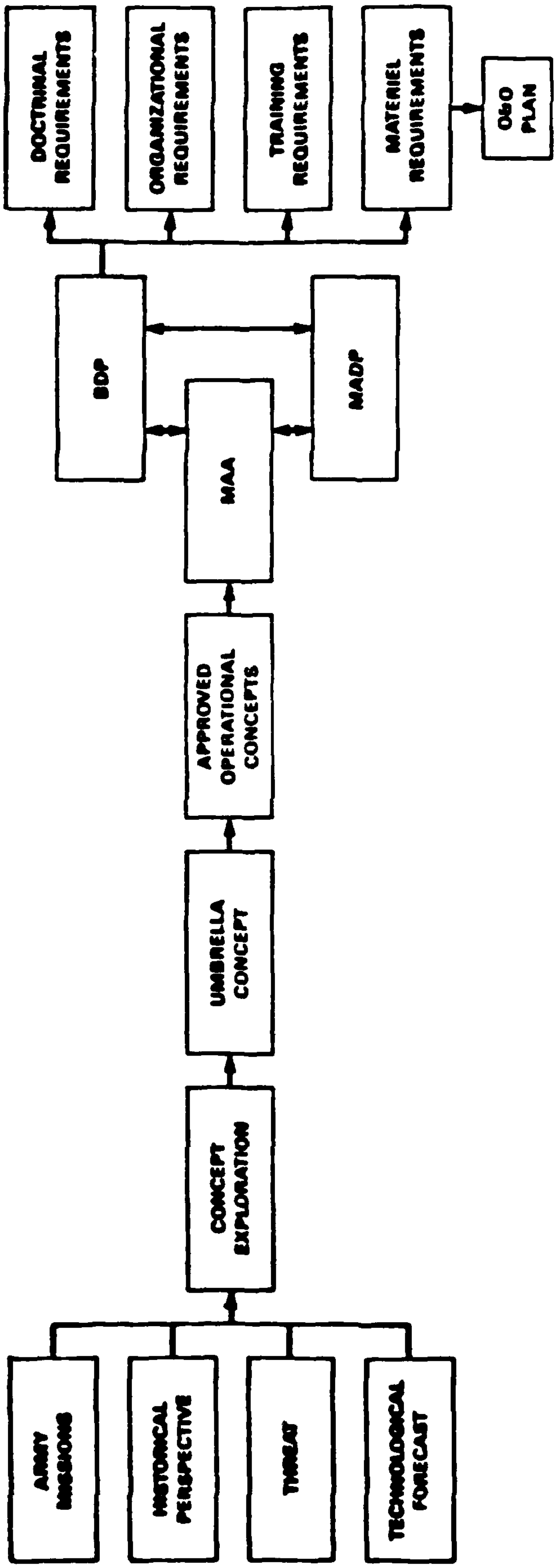


Figure 7. CBRS flow.

which are approved by the CG, TRADOC and serve as the basis for the development of the BDP.

The deficiencies and corrective actions developed in the MAA and itemized in the BDP are general in nature. A translation of these corrective actions into specific projects is required with milestone schedules suitable to support programming and budgeting decisions. The translation from MAA corrective actions to specific projects is contained in the MADP. The approval of specific corrective actions generates guidance that is disseminated to the appropriate materiel development functions. It is at this point that HQ TRADOC provides guidance to proponent schools to begin the MAP.

The New Systems Development Process

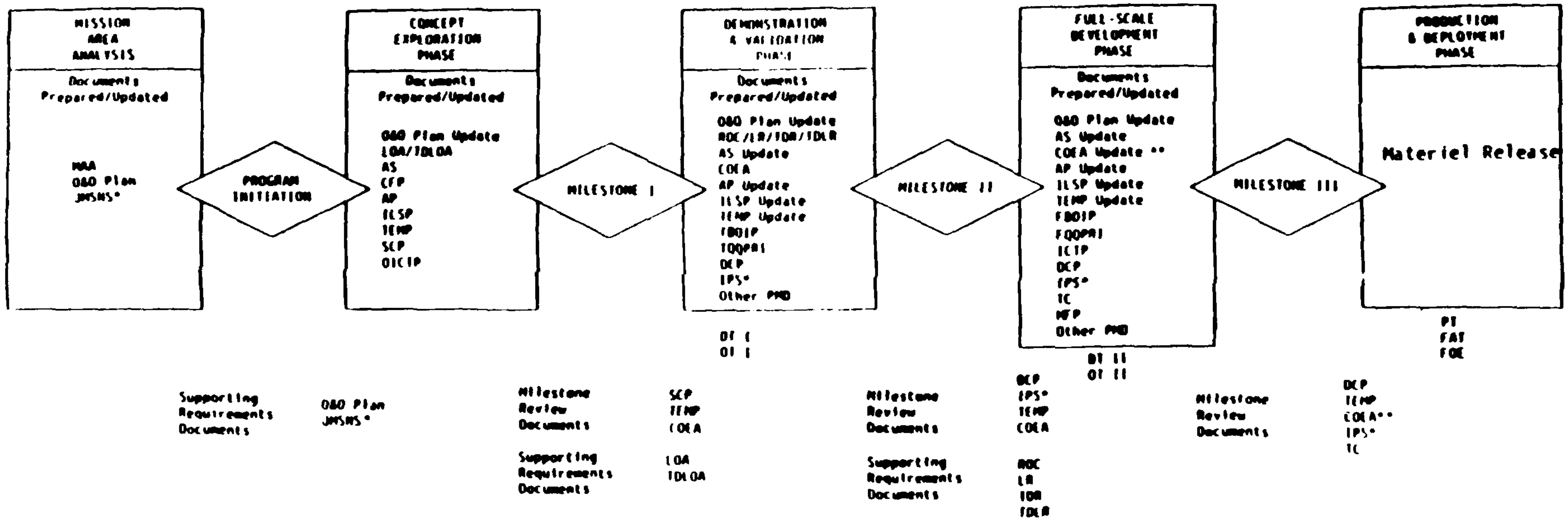
New system development encompasses an unabridged, complete materiel acquisition process and is used only when the other acquisition alternatives are unfeasible. Figure 8 provides an overview of the entire process. The documentation for the four phases following Mission Area Analysis and for the four decision points (Program Initiation and Milestone I through III) are used to describe the process. The process up to and including Milestone II is described in the following paragraphs.

Program Initiation. Initiation of a materiel acquisition program is in response to an approved requirements document. The requirements document is based on the Mission Area Analysis. For most programs, an O&O Plan developed by the Combat Developer (TRADOC) is the basis for program initiation. A Justification for Major System New Start (JMSNS) is used in place of the O&O Plan for programs whose value is expected to exceed $200 million in research, development, test and evaluation (RDTE) cost or $2 billion in procurement cost (FY85 dollars).

Appropriate FOCs provide important guidance during O&O Plan Development. In its initial stage, the O&O Plan outlines the effects of introducing a new weapon system into Army organizations. It describes how a system will be used on the battlefield and how it will interface with other systems. It also descibes the type units that will use and support the system. After milestone II the O&O Plan usually incorporates system requirements. The O&O Plan is the foundation on which Basis of Issue Plan (BOIP) and Qualitative and Quantitative Personnel Requirements Information (QQPRI), are developed. The proponent school is responsible for preparing the O&O Plan. The O&O Plan is the last stage of the CBRS process. O&O Plans are developed only for materiel solutions to MAA deficiencies.

Concept Exploration Phase. AMC and TRADOC conduct this phase of new system development based on an approved O&O Plan. This phase identifies and explores potential materiel alternatives and selects the best option for further development. Consideration of the threat, the operating environment, technical and resource constraints, and other system parameters are established through pertinent studies and the development and evaluation of experimental concepts.

THE ARMY MATERIEL ACQUISITION PROCESS



* if required

** on an exception basis

Figure 8. Milestones and phases of the materiel acquisition process.

This phase also identifies for resolution in subsequent phases, critical issues to minimize development risks. The critical issues typically include technical, operational, logistical, reliability, health and safety, manpower, training, producibility and cost concerns. Investigation of critical issues must include analysis of alternative operational and support concepts, and evaluation of manpower and logistic support resource implications of each alternative. Because of the need to explore operational and reliability issues during the concept exploration phase, the combat developer would use product one during this phase of the MAP. However, it is obvious that little data would be available to simulate or model the performance of the developing system.

The Milestone I decision review (concept selection and approval) takes place at the end of the Concept Exploration Phase. The decision validates the requirement and approves the aquisition strategy proposed by the developer to satisfy the materiel requirement.

Demonstration and Validation (D and V) Phase. This phase consists of steps necessary to verify preliminary design and engineering, accomplish necessary planning, analyze trade-off proposals, resolve or minimize logistic and reliability problems identified during the Concept Exploration Phase, and validate the concept for entry into the Full-Scale Development (FSD) Phase. This phase is conducted by AMC, in coordination with TRADOC, based on direction provided in the SADM. In addition, during this phase and before Milestone II, TRADOC, in coordination with AMC, prepares as a requirements document, either a Required Operational Capability (ROC), Letter Requirements (LR) (for low value items), or equivalent requirements document for training devices. Both documents contain essentially the same information. They both state concisely the minimum essential operational, technical, personnel, manpower, safety, health, human factors engineering, training, logistics, and cost information necessary to initiate the Full Scale Development Phase, or the procurement of the materiel system. These documents support Full Scale Development (6.4), or may be used to support acquisition of nondevelopment items (NDI's).

The Current Process of Developing Objectives, Criteria and Conditions

The current process of developing requirements for a new system is usually begun during the concept exploration phase of the MAP. The process is not reliable because it is situation dependent and consists of various strategies applied at the discretion of each combat developer. Most often the approach relies on the unsystematically collected opinion of one or a few experts.

Other strategies include the use of the predecessor system's requirements increased by a significant amount; for example, 10%. Such a strategy is evidenced in a requirement such as "deliver 10% more rounds on target per unit of time." Obviously such requirements are not based on the deficiency specified in the MAA; neither are they based on the performance required to ameliorate the deficiency. On the contrary, such requirements are often based on simply arbitrary decisions.

Requirements for a new system often embody the capabilities of a new technology irrespective of the deficiency they are supposed to address. In such cases, new technological advances simply become the requirements for a new system regardless of the specific materiel deficiencies the Army may have. In such a case the requirements for a new system circumvent the purpose of the CBRS.

Problems with the Current Process

One obvious problem with the current process is the lack of a systematically applied, concrete methodology with which to develop early requirements for a developing system. The lack of such a methodology often makes the combat developers' job more difficult, vague and labor intensive. Without a simple concrete methodology, the combat developer has no small number of easy and discrete steps to follow.

Developing requirements without a systematic and concrete methodology may also result in criteria and objectives that reflect the biases of those who develop them. Moreover, frequent personnel turnover usually occurs in combat development divisions. This means that requirements unsystematically developed from expert opinion or a complex methodology will lack a good audit trail. Thus the next step which needs to be completed for those requirements only partially developed will not be obvious. Hence, new combat developers taking over the development process for someone who has transferred will not know where to continue with the requirements development process.

Similarly, if the specifications of a technological advance are used as the requirements for a new system instead of requirements specifically designed to ameliorate a deficiency described in an MAA, other problems will result. These problems would primarily entail the purchase and/or development of an unneeded system or a system with unneeded capabilities and/or components.

The User

The combat developers in the TRADOC schools and centers are the typical developers of requirements for a new system and thus are the probable users of product one. These personnel are both active duty officers and civilian personnel. The following description of these personnel is based on interviews conducted over the past year with 12 of them at six schools. In addition, two retired combat developers were interviewed expressly for this paper.

The most typical ranks of the officers who are combat developers are captain and major. The combat developments directorate at each school or center is typically headed by a colonel. In addition, each directorate usually contains at least one lieutenant colonel.

Each officer assigned to combat developments usually serves a tour of two or three years. Typically officers begin their tours in combat developments with very little if any relevant experience in developing requirements for new systems. Gaining enough experience to adequately perform their job assignments usually takes about six months. Thus officers will only be sufficiently experienced to adequately perform their jobs in combat developments for 75% of their tours (18 months of a 24 month tour). This means that many combat developers will be transferred to a new duty position while they are in the middle of developing the requirements for a new system. Because of this the processor must be easy to learn by a new combat developer.

Civilian combat developers usually remain in their positions for longer periods than the officers. However, there is moderate personnel turnover also among the civilians.

Aside from their tenure in combat developments, the military and civilian personnel are very similar in terms of user characteristics. Most of the military and the civilian combat developers have similar training, education and capabilities. Their skills are typically "non-technical" and their job duties do not usually call for technical skills such as modelling. Most combat developers are not capable of conducting or interpreting any inferential statistics including any tests associated with regression analysis. Mostly they follow non-technical procedures when developing new system requirements and sometimes they rely on the results of tests or analyses done by others. Thus, product one must not require technical skills (e.g., regression analysis or modelling).

Another reason product one must be easy to use is that the combat developers who might apply it are overworked. Typically, they are responsible for several new systems and other additional duties as well. They do not have much time to either learn a new methodology or apply one. Thus product one must not be labor-intensive nor difficult to learn.

The Requirements for Product One

There are several theoretical and practical requirements which should guide development of product one. The theoretical requirements pertain mostly to the development of objectives and criteria. The practical requirements stem primarily from the problem, its context and the requirements related to the characteristics of the user.

Given the problem of developing performance requirements, its context and the user, the Army needs a user friendly methodology that will assist the combat developer in systematically developing requirements for new systems. Consequently the whole methodology must be one that will:

1. Be reliable and valid.
2. Derive performance-based requirements from MAA deficiencies.
3. Not be heavily data-dependent in the early stages.
4. Be simple to learn and apply.
5. Be computer-based.

Each of the five requirements is explained in detail in the following paragraphs. The five requirements will be referenced by number in the following descriptions of the processor.

In order for product one to help produce valid requirements it must be a reliable methodology and thus be a systematic methodology. This follows from the fact that the validity of any criterion or score is limited by its reliability. In statistical terms, the upper potential limit of the validity of any score is the square root of its reliability (Nunnally, 1978).

The fact that the tours of duty for the military combat developers are often only two years in duration has implications for the reliability of the methodology. The duration of the tours of duty places extra emphasis on the reliability of the methodology. Since it is probable that many combat developers will leave the development of requirements somewhere in the middle of their development, the procedures they use need to be extremely explicit and of a step-by-step nature (i.e., reliable). Thus with a very proceduralized processor, a new combat developer taking over the development of requirements already begun could pick up the development process and see exactly the steps that were already performed and those remaining that need to be performed. In contrast, using a methodology that is complex and whose steps are not rigorously spelled out would result in the new combat developer not knowing exactly what had already been done and more importantly not knowing which is the next series of steps to take.

MAAs are designed to identify deficiencies related to unit missions. Thus deficiencies are supposed to be performance-related. Logically then, the earliest requirements for a new system should be performance-based requirements. They should be statements of what the system should be able to do to ameliorate the deficiency.

Early in the new system requirements development process, few relevant data are available from which to derive requirements for a new system. Thus a requirements development methodology which is heavily data-dependent in its early stages is doomed to frustrate the combat developer. In addition, such a methodology is probably going to result in the development of requirements being delayed beyond their earliest useful date.

Military combat developers of new system requirements usually begin their tours of duty without the job experience required to immediately develop adequate new system requirements. Moreover, they often have heavy workloads in the form of several systems to develop and other duties to perform. In addition their technical skills are few and they have already seen too many methodologies requiring the memorization and use of weighty user's manuals. Hence, product one must be a system that is easy to learn and apply. Product one will not be used by combat developers if it requires a lengthy series of steps that are not easy or that are labor intensive.

Similarly, product one should be a computer based system with embedded training that requires little if any spin-up time. The user should be to start using the product one immediately. Combat developers will not be willing to spend weeks learning how to use a new tool that is supposed to make their jobs easier.

In addition to the requirements already mentioned which are derived from the general problem, there are several requirements the methodology must meet to be a good measurement system and thus be psychometrically sound. These types of requirements are explained in the following paragraphs.

The MAA identifies performance deficiencies and the type of system and performance that would be required to obviate the deficiency. However, the MAA does not specify a rule or a standard by which a judgement can be made about the performance which is required. Criteria are rules or standards for making judgements. However, in order to serve as rules or standards, criteria must be reliable, practical and not trivial or biased (Smith, 1978).

To be reliable a criterion must produce the same results or decision when used by different personnel in the same situation. For example, two combat developers both independently applying a criterion to the evaluation of a new system should both appraise the system to the same degree of effectiveness. Thus in most cases criteria must surely be quantitative in order to produce reliable results. To be practical a criterion must be easily applied and not require an inordinate expenditure of resources to apply.

Another aspect of practicality is the sensitivity of a criterion. Criteria should be sensitive but only as sensitive as the situation warrants. For example, if a potential early system performance criterion is focused on the traveling speed of the new system, focusing on increments of a single mile per hour would probably be too small a unit of change to measure.

The development of criteria must also address the "single criterion" problem which subsumes the issue of how many criteria should there be or is the group of criteria complete. With several criteria which product one will have to produce, a related problem concerns the relationships between the criteria or their relative weights. A similar question focuses on whether the criteria should have compensatory relationships with each other such that tradeoffs are possible. The answer to the latter depends on the criteria themselves and their ultimate purpose. However, given the necessity for product one to be capable of producing multiple criteria for each new system and the intense competition of many new product developments, it would be very useful if not absolutely necessary that product one produce relative weights for each of the criteria it helps identify. Relative weights for each of the criteria would help in making a decision between two alternatives to a materiel deficiency competing during the Demonstration and Validation Phase of the MAP.

There is a long standing consensus that relevancy to important goals is the most important aspect of a criterion (Kendal, 1956; Guion, 1961). This means that criteria used to evaluate the performance of a new system should be very relevant to the important goals or objectives of the new system. And what are the requirements for the objectives of a new system? One avenue from which to determine the requirements for objectives is to draw on the voluminous literature on the evaluation of work performance or performance appraisal. Work performance is typically evaluated by measuring how well people accomplish their goals or objectives. This is especially the case with senior personnel not assigned repetitive production tasks. Inherent in the problem of developing objectives and criteria for the performance appraisal of senior personnel is the need to tie their objectives to the goals of their organization (Campbell et al., 1970). This need logically follows from the fact that senior personnel are more clearly hired to foster the higher level goals of an organization; whereas, for example, production type workers are hired to accomplish tasks less clearly related to an organization's higher level goals.

Clearly there is a parallel requirement for important systems of an organization. Important systems are similar to senior personnel - both are important, few in number and obtained to directly foster at least some of the larger or higher level goals of an organization. Thus, we should evaluate important systems in terms of how well they contribute to the accomplishment of their organization's goals. The goal of an Army unit is the accomplishment of its mission.

The same argument for the evaluation of new systems in terms of their contribution to unit effectiveness (accomplishment of the unit's mission) can be seen in the fact that new systems are supposed to be developed and procured to ameliorate deficiences. MAA deficiences are descriptions of a unit's less than adequate capability to perform its mission(s). Thus new systems are supposed to be built to help units accomplish their mission. Therefore new systems should be evaluated in terms of how well they help units accommplish their mission.

Another fundamental aspect of evaluating a new system concerns the use of the entire unit's overall performance. While new systems may be built to help ameliorate a problem a unit has with accomplishing <u>part</u> of its mission, evaluations of a new system must use the overall performance of the unit as the ultimate criterion for at least two reasons. One reason is that while a unit may have had a deficiency with regard to a part of its mission or one of its functions, the introduction of a new system may have negative synergistic effects on the whole unit. The new system may cause previously successfully performed parts of the unit's mission to now be performed less than successfully. Thus the effect of the new system on its whole unit must be assessed and used as the ultimate criterion for judging the performance of the system.

Another reason for using the overall performance of the unit as the ultimate criterion for system effectiveness is the fact that it is probably not feasible to reliably tease out the effect of a new system on only one or a few of the functions of a unit. There is too much interaction between functions. A much more valid approach is to look at the effect of the new system on the overall performance of the unit.

Since criteria need to be relevant to the objectives of the new system, obviously the required classes of performance must also be relevant to the objectives. Thus the classes of performance must stem directly from the objectives and thus also be stated in performance terms. The classes of performance and criteria are further specifications of the objectives which are derived from the unit's mission.

The preceeding discussion of the requirements for objectives and criteria has identifieo some fundamental psychometric requirements which are requirements for product one in addition to the five described earlier:

6. Criteria must be reliable and practical.

7. Relative weights must be established for the criteria.

8. The classes of performance and criteria must be relevant to important goals (objectives) of the system.

9. The objectives of the system must embody the goals of the unit for which the system is being designed. (The goals of the unit are embodied in its mission.)

10. Thus the system should be evaluated in terms of how well it contributes to its unit's performance.

11. The entire or overall performance of the unit should be the ultimate criterion for evaluating the performance of a new system. (This is the case because the introduction of a new system into a unit could affect any part of the unit's performance).